NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

REQUIREMENTS ANALYSIS AND ARCHITECTURAL DESIGN OF A WEB-BASED INTEGRATED WEAPONS OF MASS DESTRUCTION TOOLSET

by

Richard B. Jones

June 2004

Thesis Advisor:	Mantak Shing
Thesis Co-Advisor:	Doron Drusinsky

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Requirements Analysis and Architectural Design of a Web-based Integrated Weapons of Mass Destruction Toolset	**5. FUNDING NUMBERS**
6. AUTHOR(S) Richard B. Jones, Major	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

In 1991, shortly after the combat portion of the Gulf War, key military and government leaders identified an urgent requirement for an accurate on-site tool for analysis of chemical, biological, and nuclear hazards. Defense Nuclear Agency (now Defense Threat Reduction Agency, DTRA) was tasked with the responsibility to develop a software tool to address the requirement. Based on extensive technical background, DTRA developed the Hazard Prediction Assessment Capability (HPAC). For over a decade HPAC addressed the users requirements through on-site training, exercise support and operational reachback. During this period the HPAC code was iteratively improved, but the basic architecture remained constant until 2002. In 2002, when the core requirements of the users started to evolve into more net-centric applications, DTRA began to investigate the potential of modifying their core capability into a new design architecture. This thesis documents the requirements, analysis, and architectural design of the newly prototyped architecture, Integrated Weapons of Mass Destruction Toolset (IWMDT). The primary goal of the IWMDT effort is to provide accessible, visible and shared data through shared information resources and tem plated assessments of CBRNE scenarios. This effort integrates a collection of computational capabilities as server components accessible through a web interface. Using the results from this thesis, DTRA developed a prototype of the IWMDT software. Lessons learned from the prototype and suggestions for follow-on work are presented in the thesis.

14. SUBJECT TERMS WME,WMD, Chemical, Biological, Nuclear, hazard prediction, software tools, distributed, client-server, SOAP, XML, JAVA, DTRA, HPAC, IMEA, INCA, targeting, web-services,J2EE, IWMDT	**15. NUMBER OF PAGES** 119
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

REQUIREMENTS ANALYSIS AND ARCHITECTURAL DESIGN OF WEB-BASED INTEGRATED WEAPONS OF MASS DESTRUCTION TOOLSET

Richard B. Jones
Major, United States Army
B.A., Ohio University, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
JUNE 2004

Author: Richard B. Jones

Approved by: Mantak Shing
Thesis Advisor

Doron Drusinsky
Thesis Co-Advisor

Peter Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In 1991, shortly after the combat portion of the Gulf War, key military and government leaders identified an urgent requirement for an accurate on-site tool for analysis of chemical, biological, and nuclear hazards. Defense Nuclear Agency (now Defense Threat Reduction Agency, DTRA) was tasked with the responsibility to develop a software tool to address the requirement. Based on extensive technical background, DTRA developed the Hazard Prediction Assessment Capability (HPAC). For over a decade HPAC addressed the users requirements through on-site training, exercise support and operational reachback. During this period the HPAC code was iteratively improved, but the basic architecture remained constant until 2002. In 2002, when the core requirements of the users started to evolve into more net-centric applications, DTRA began to investigate the potential of modifying their core capability into a new design architecture. This thesis documents the requirements, analysis, and architectural design of the newly prototyped architecture, Integrated Weapons of Mass Destruction Toolset (IWMDT). The primary goal of the IWMDT effort is to provide accessible, visible and shared data through shared information resources and tem plated assessments of CBRNE scenarios. This effort integrates a collection of computational capabilities as server components accessible through a web interface. Using the results from this thesis, DTRA developed a prototype of the IWMDT software. Lessons learned from the prototype and suggestions for follow-on work are presented in the thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I wish to thank my three direct supervisors, Mark Ferrell, Dave Myers and Walt Zimmers for their encouragement, support and leadership over the period of study for completion of this Master's degree. Over the last four years, I deployed over 300 days to foreign nations, served as the program manager for six different multi-year, multi-million dollar programs, a feat that was possible because of your support and leadership. Your patience and mentoring allowed me to apply the knowledge gained in my studies immediately to the roles you allowed me to serve in within your branches. Thank you for your patience and support.

Though given the support and encouragement of my leaders, the ability to withstand the academic rigor would not have been possible without the continued guidance of my advisor, Mantak Shing. Four years ago an inexperienced Army Major sought an advanced degree in Software Engineering, today an experienced Software Engineer looks back and appreciates the tremendous support and knowledge that you shared. Of special note for consistently professional and current and relevant advice are James Michael, Richard Riehle and Bill Ray, each you epitomized the excellence in education and engineering that the program strives to demonstrate to the student body. Thank you for your time and your willingness to work with distant learning students in a manner that served the students academic and professional objectives.

And most importantly I want to publicly thank my wife Michelle and my daughter Jessica for years of support and encouragement in my studies. Working by day and studying by night coupled with such a high number of days of deployment would have caused me to quit had it not been for you guys. Of the many blessings that GOD has given me, you two are the greatest gifts. As proud as I am to achieve this degree, and as humbled as I am to be called a Software Engineer, it pales when compared to being called your husband and father.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

More than a decade before the 9-11 terrorist attack, the Coalition Forces serving in Iraq and Kuwait were faced with the potentially deadly uncertainties of Weapons of Mass Destruction (WMD) and environmental hazards. The exact threat after the initial days of the Gulf War was unknown and the demand for immediate and accurate assessments of the potential of WMD threats was imminent. A reliable capability was desperately needed for providing near-real time predictions of the hazardous materials present in Iraq and neighboring countries. Operation Desert Storm accentuated the need for an automated hazard predication tools on-site for use by the decision maker with rapid and acceptably accurate results. Given the lack of tools in theater that accounted for terrain, weather, agent and delivery, the standard of acceptable accuracy was not very strict. The commanders needed a tool that would approximate the general exposure to hazardous materials using contours of threat as opposed to the current generalized area assessment available through the ATP-45 triangle method. Best available data became a mantra that resulted in a very general plot that rarely facilitated the effective evaluation of course of action or decision-making involving prediction of collateral damage.

During the military campaign and in the aftermath during the cleanup, predictions of the collateral effects of potential WMD and environmental hazards were inefficient and untimely. Without an integrated tool on-site, hazard prediction analysis was conducted through a slow and inconsistent reachback process conducted at the Defense Nuclear Agency (a predecessor organization to the Defense Threat Reduction Agency (DTRA)). Analysis was performed inconsistently depending on the skills and techniques of the analyst on shift. The forward units sent request for information (RFI) back to the DTRA Headquarters, and then in most cases were forced to wait up to 12 hours for the results to be sent back to Theater. And, when the results were returned quite often the results did not answer the nuances of the emergent situation.

This experience prompted DTRA to develop a collection of tools for on-site targeting, consequence assessment, hazard prediction and collateral effects. The tools were designed to operate as stand-alone applications, each loaded on the local machine with databases locally managed. There were two primary tools developed by DTRA that addressed these requirements from 1991-1996: Munitions Effects Assessment (MEA) and Hazard Prediction and Assessment Capability (HPAC). MEA was developed under a Counter-Proliferation Advanced Concept and Testing Demonstration (CP ACTD) effort, designed as a targeting tool to provide attack and collateral effects predictions. HPAC was developed to provide prediction data on hazardous release of chemical, biological and nuclear incidents, using weather, terrain, agent, and transport data. The Initial focus addressed the shortcomings of the recent conflict as identified by a post-conflict study conducted by IDA shortly after the Gulf War. An excerpt of the DIA report stated[1]:

> .... continued concern about the inability to describe the many variables of the agent-munition release mechanism. The panel agrees with the CIA that "huge uncertainties remain" in the number of rockets present for destruction and the number of those rockets destroyed. Among the other major variables for which there remains much uncertainty are total quantity of agent released, mechanism of release, and purity of agent.

B. THESIS ORGANIZATION

This thesis outlines the author's initial design and research and then subsequent efforts by DTRA that lead to the development of the "Integrated Weapons of Mass Destruction Toolset" (IWMDT). This thesis investigates and reports on the software effort and the associated software systems engineering implications.

This chapter gave a brief introduction to the problem and the motivation of the research. Chapter II presents an overview of the supporting technologies. Chapter III

[1] Agent Purity comment – IDA report DefenseLINK News Release, Reference Number 681-96, 20 December 1996.

documents the system requirements and Chapter IV describes the architectural design of the IWMDT software. Chapter V discusses the requirements for configuration management, validation, verification, and accreditation, as well as system and data security for the IWMDT software. Chapter VI contains the lessons learned from the IWMDT prototype, a conclusion, and recommendations on future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. ANALYSIS OF TECHNOLOGY

A. ANALYSIS OF WAR HAZARD ASSESSMENT TECHNOLOGY

Meeting the immediate Chemical, Biological, Radiological, and Nuclear Effects (CBRNE) needs of the warfighter, and establishing methods for applying technology to meet their requirements became an important goal for DTRA after the Gulf War of 1991. During the war, there was a viable threat of CBRNE casualties due to Saddam Hussein's previous use of such weapons[2]. Without an automated hazard prediction capability and integrated targeting system, the only method that commanders were able to employ to determine threat areas and threat levels was through the use of a generalized CBRNE threat triangle. The triangle referred to as an ATP45 prediction[3], accounts for direction and predicted length of a CBRNE threat. Because the manual triangle method is designed for gross area prediction it is limited in detailed assessments affecting specific areas of operations, the specificity of the data is critical for making decisions that potentially will impact locals, enemy forces and your own forces on the battlefield.

The identified requirement for an on-site analysis capability led to the development of HPAC in 1991. Drawing from over fifty years of CBRNE excellence and scientific research[4], DTRA transformed research projects and scientific studies into a software tool to meet these critical requirements. In 1991, after the Gulf War, lessons learned and formal reviews identified requirements and technology gaps that needed immediate attention. DTRA developed HPAC to address the technology gaps that left our troops exposed to potential CBRNE threats during and immediately after the Gulf War. Using this tool allowed commanders to more accurately predict collateral damage and hazard area predictions, and provide greater safety for the troops.

---

2 http://www.defenselink.mil/news/Jan2003/n01232003_200301234.html, May 2004

3 http://www.globalsecurity.org/military/library/policy/army/fm/6-20-30/Ref.htm, May 2004

4 http://www.dtra.mil, May 2000

From 1991 to 1999, DTRA continuously improved the HPAC tool, issuing releases approximately every six months. Gradually the code was re-engineered leading to a major paradigm shift in 1999. In 1999, DTRA made a significant architecture modification by developing a design based on new distributed object architecture methods (Figure 1). In a paper published by Ronald W. Lee in Aug 2002[5] the HPAC architecture is described as having several system level requirements, these requirements are valuable supporting technology improvement that led to the IWMDT concept. Mr. Lee's general outline explains the attributes of the HPAC structure as: *portability* - platform independence, *extensibility* - allow addition of new capabilities to a deployed system, deployment - *flexibility* for a range of situations and environments, support - *client-server operation*, and expedited *integration* of HPAC components in other systems. More detailed analysis of HPAC and the IMEA tool introduced next are presented later in Chapter IV of the thesis.

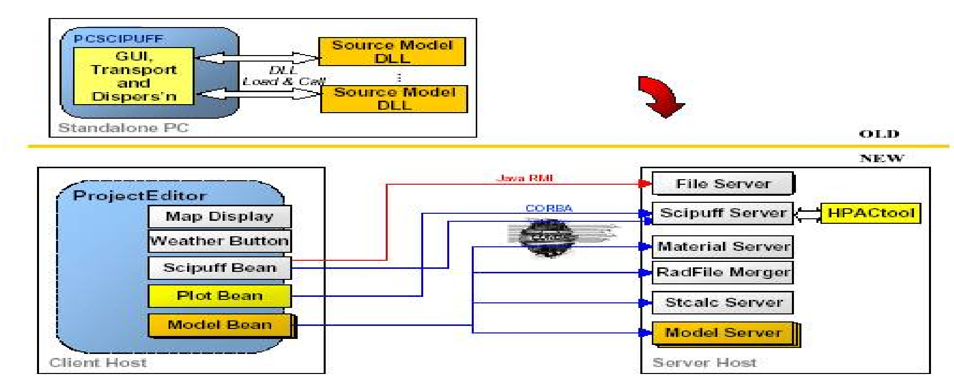


Figure 1. Redesign of HPAC (ORNL – 1999.)

B. ANALYSIS OF TARGET ASSESSMENT TECHNOLOGY

While HPAC addresses the dispersion and transport of agents, Munitions Effectiveness Assessment (MEA) integrates the dispersion with the structural and weapon target dynamics of targeting.

5 http://wigner.cped.ornl.gov/HPAC/info/paper.tm.pdf

The United States European Command (EUCOM) under the Department of Defense's (DoD) Counterproliferation Advanced Concept Technology Demonstration (CP ACTD) commissioned MEA. Specifically designed for deliberate and crisis planning, MEA allows users to design three-dimensional models of above- and below-ground facilities and simulate end-to-end attacks using precision guided air-to-ground and other types of munitions of conventional and WME sites. Though initially designed to assist warfighters in visualizing and weaponeering fixed targets, the tool was later integrated with HPAC for an integrated attack/dispersion solution. The new integrated application was entitled, Integrated Munitions Effectiveness Assessment (IMEA). Most target types can be specified to contain WMD, in those cases any resulting expulsion is passed to HPAC for transport and dispersion. Later in Section IV.G of this thesis a use case and description of information exchange will demonstrate this process.

IMEA uses fast running, physics based algorithms to predict optimal weaponeering solution. Resulting cratering, fragmenting, blast damage and collateral effects are presented to the user as probability and extent of effect solutions. The IMEA application is quite complex and provides a very high degree of flexibility in design of targets and sites. Using a hierarchical data structure, presented to the user in a tree structure as shown in Figure 2, the application and the user maintain a consistent and traceable task flow.

The arrows point out the use of nodes and tree structures for databases. Arrow 1 is the top-level "database" level, arrow 2 is the "site" level, arrow 3 is the "target" level and arrow 4 is the "model". Through the use of this hierarchical structure the user is able to easily associate targets to sites and allow groupings of targets.

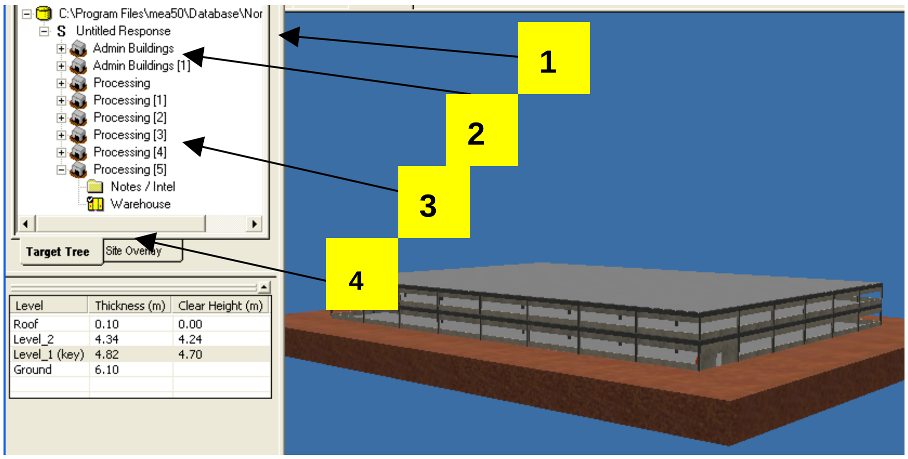


Figure 2. IMEA Screen Shot.

C. ANALYSIS OF DEVELOPMENT METHODOLOGIES

Developing tools in the 1990s was considerably different than developing tools in the 2000s. This difference is largely due to the application of technologies and learned experiences that have improved the engineering of the source code and the exchange methods. So it follows that tools developed in the 1990s and reengineered in the 2000s are often more complicated than newer tools with pure 2000 technology and design methods applied. But, the transformation of capabilities from the 1990s versions of IMEA and HPAC to the 2000 architecture is being accomplished much smoother than would be expected. Much of the success is due to the consistent requirements, the unparalleled cooperation of over five companies, and strong leadership. Before we talk about the specific methods implemented by the team it is wise to define a few processes and efforts that impacted this process.

1. Evolution And Revolution Of Technology

When developing innovative technologies, the methods and processes are often as innovative as the product that results from the effort. Though there are clearly established disciplines that are followed, it is the application of the gray areas that separate the mundane from the fantastic. Because innovations are by nature unique, even the “experts” do not agree on the best path to follow to create a new capability. Some say the solution to solving hotly contested concepts such as band-width limits, security limits, concurrent processing, shared common data etc. is to invoke evolution; others say we need a revolution. But what is the difference in the two, and why do we care?

If the goal of the technology transformation is to exercise a gradual process in which something changes into a different and usually more complex or better form, then choosing **evolution** is most appropriate. If the goal is to find a drastic and far-reaching change in the way of thinking and behaving, then opting for a **revolution** is more appropriate. Though the definitions of these words leave the reader with a “who cares” attitude the effect of the choice is actually quite serious. Choosing to revolutionize an industry is usually painful, long-term and costly. On the other hand the process of slow and gradual change is not a good choice if the industry influences human lives and failure to act boldly may lead to unnecessary suffering and/or death. Additionally many times the slow process can result in higher costs and less directed results due to the longer period of performance and the loss of momentum when lasting longer periods.

In the case of software development providing CBRNE assessments, the industry borrowed time because the probability of large-scale CBRNE attack was considered low prior to 1991. After the Gulf War, the author discovered that DTRA choose to invoke a hybrid of both revolution and evolution. Fifty years of experience clearly warned DTRA of the impact of pre-mature implementation of tools, but current demands necessitated immediate development to meet the needs identified in the War. Because of unique situations, DTRA had the two spirals of change acting in harmony to allow internal revolution in support of evolution goals and internal evolution in support of revolution goals.

Regardless of the spiral method of evolution or revolution, the goal is to produce a product without unduly affecting the user operational tempo and yet providing near real time assessments. From 1991 to 1999, HPAC and IMEA provided a core prediction capability and targeting solution that is best characterized as a **revolution** in CBRNE prediction. From 1999 to 2003, HPAC development could be best characterized as an **evolution**, as it evolved into an open architecture. By allowing more modular development and incremental module and functional improvements, the capability was improved and the user base expanded. Starting in 2003, a **revolution** once again occurred as new technology made web-service architectures possible. The new approach to delivering accessible and visible data provided a much greater level of interoperability and rapid collaboration of users, decision makers and subject matter experts (SMEs).

A common understanding of the processes is a good start, but there are still many issues that need to be resolved before this revolution becomes solid enough for widespread use. The current transformation is analogous to the 1990s effort from primarily mainframe architectures to the client/server architectures so prevalent in the last decade. The migration from the client/server methods of the 1990s to the use of web-services in 2004 is worthy of a short discussion as it lays a foundation for the migration of HPAC and IMEA to the current tool IWMDT.

Just as the vision of lowering the costs from the mainframe to the client/server was a noble yet naive vision, history may repeat itself in the web-services area. In the 1990s scenario, the cost for MIPS (millions of instructions per second) was lowered, but the overhead associated with the maintenance of the high count of machines was increased. Additionally a bevy of applications were released in the maturing of the client/server movement that further fractured the industry and complicated the integration of information and processes that were critical to the user's success. Once again as we watch the proliferation of web based services, a whole plethora of services will be offered to meet the needs of a naive user market anxious to be revolutionist and to gain an advantage over competitors. Which companies will make the wise choice and what will the right choice be?

The good news is that the standard protocols are gaining support from industrial leaders such as Microsoft and Oracle. The bad news is that though many of the tools and services are common, the implementation of many other services is still evolving. This evolution is sure to cause implementation issues as web-services are created on one vendor’s tools and then deployed on another vendor’s hardware.

This uncharted sea of technology will either be circumnavigated as the client/server template suggests with a glut of providers competing for the survival of the fittest, or a new paradigm will emerge. Signs are positive that a new paradigm might prevail. This new paradigm will be characterized by widespread contractor teaming, and shared data and application processes across common information grids. This paradigm has been active and improving at DTRA for the last two years in the IWMDT program.

Pete Lindstrom, Director of Security Strategies at the Framingham, Mass.-based Hurwitz Group, says, "In a lot of ways, web-services are all about the interoperability problems we have today between systems," he points out. Thus, it's only natural that at first there would be these kinds of issues. Respectfully, the author offers that the real issue is not the interoperability of the software and hardware, but the competitive environment that capitalism creates. In the early to late 1980s and continuing into the late 1990s, the government encouraged lengthy competitive trials that led to a “survival of the fittest” marketplace.

This process was intended to provide the government the best product at a competitive price, but the process led to costs rising to the contractors. This caused the cost of each of the contracts to rise in an effort to meet the overhead associated with preparing the proposals and in some cases delivering elaborate prototypes. This method encouraged the contractors to work against each other instead of with each other, and the government paid the price for their competition.

In the early 2000s a very positive trend is prevailing at DTRA, this trend involves very complicated partnering and collaborative efforts between competitors. If this trend continues the norm will include closely integrated solutions and the anomaly will be “survival of the fittest”. The anticipation will mount as the industry decides if they will follow the trend of government contractors, or develop a less effective open bid process that has proven ineffective. By partnering the contractors in an Indefinite Quantity Indefinite Delivery (IDIQ) contract each of the companies is assured a certain amount of market share, and the incentive to “cooperate and graduate” is greatly improved.

D. INTEGRATION OF TECHNOLOGY AND REQUIREMENTS

Mid-year 2000, the author arrived at DTRA. Eager to serve and yet limited in his skills and background, contracting and software engineering education was immediately necessary. This education came through the Naval Postgraduate Software Engineering Masters Degree program and on the job training as program manager and government lead on three contracts. The author mentions these two influences because without the coordinated occurrence of these two influences many of the programs and events that occurred in DTRA from 2000-2004 would not have been possible.

Daily across the government, military officers are given the task of serving as Contracting Office Technical Representative (COTR) with little or no formal training. This responsibility can be quite challenging to the school trained program manager, to the novice it can be overwhelming. When the author was assigned this task, he was the later. And as such, he had nothing to guide his decisions except common sense and ethical behavior. This was noble but hardly the stuff that successful COTRs are made of. Further complicating this task were the many other responsibilities that are expected of military officers. If it were not for the excellent instruction and practical application that NPS offers in the distant learning program, the task may have been too large. But, able to draw on the professional knowledge of the NPS staff, the author charged the hill of ignorance with confidence, constantly improving his knowledge and applying the classroom instruction.

Beginning in 2000, the author had two roles at DTRA. One role was as a program manager and software engineer, the other was as a warfighter and a CBRNE SME in direct support to Combatant Commands. As a military warfighter the responsibility to provide commanders rapid, accurate and consistent CBRNE recommendations was a crucial part of his role. Deploying over 300 days in direct support of Combat Headquarters meant that decisions made as a program manager and engineer directly affected his ability to accomplish a wartime mission with the same tools he was developing.

As the senior military CBRNE software engineer in DTRA, the author deployed to Central Command (CENTCOM), United States Forces Korea (USFK), Supreme Headquarters Allied Power Europe (SHAPE), European Command (EUCOM) and Pacific Command (PACOM) two to three times each annually. On these deployments it was the responsibility of the DTRA SME to integrate CBRNE predictions and targeting planning into existing staff operations. DTRA tools were able to describe specific areas that contained levels of hazard, but were not able to tie it specifically to units or other systems within the command posts or simulation. This lack of integrated data would remain an issue until the 2003 development of HPAC-J, discussed later in Section IV.C of the thesis.

Considering the limitations discussed above, the data that DTRA was able to provide was still a magnitude of quality better than other tools available. In addition, a more concerning immediate limitation was the deployment requirements. The tool accounted for so many factors, including terrain, weather, feature data, etc that the hard drive space requirements were in excess of 1GB. The large storage requirements and the requirement for administrator level access to load the software often were too demanding for the users to accept.

These requirements ranged from burdensome at some commands to too extensive to allow DTRA to load HPAC at other sites. As the tool improved the local installation and operation requirements continued to escalate until in 2001 users demanded another tool deployment option. The only option for deployment of the tool and CBRNE prediction capability was to continue the heavy burden on the commands or develop a system that was broadly accessible, consistently visible and that had very limited local requirements.

In 2002, the author developed a software requirement document to develop a web-browser capability for HPAC to address the reduced requirement option. The original intent was to allow users to log on to a standard web-browser and access HPAC capabilities. After the initial intent was articulated, new technologies began to change the methods for providing net-centric data. The new paradigms was ideal for meeting the need of providing the SME tools and yet reducing the burden on the users. Continued work in this area led the author and DTRA leadership to investigate the application of net-centric processes to assist in this transformation. Later in this thesis we examine some of the services, architectures, processes, and integrated tools that have impacted on the original vision and the IWMDT development effort.

E. PARADIGM SHIFT – NET-CENTRIC FUSION

Major paradigm shifts across broad communities usually result in large delays, major funding shortfalls and confusion until they mature. In the early 2000s a paradigm shift occurred across the Department of Defense, which focused on the application of web-services to fuse data and processes. This shift heralded by many over the last ten years is still evolving, but the initial shifts were smooth enough that the community managed it successfully. One of the recent applications of this concept, which benefits IWMDT, is the Horizontal Fusion (HF) Portfolio. In Chapter VI “Discussion and Conclusion” of this thesis, this technology application is more closely examined.

IWMDT is ensuring that all program decisions support the integration into the Horizontal Fusion initiative, which is a cornerstone to the Secretary of Defense vision of Force Transformation. Secretary Rumsfield’s vision – to “think differently and develop the kinds of forces and capabilities that can adapt quickly to new challenges and to unexpected circumstances”, clearly imputes the urgency that the Department has to ensure warfighters and analyst have timely and accurate data. The key to this effort is not only the provision of data but also the method with which that data is handled and provided to communities of interest.

The realization that providing processed data from subject matter experts to commanders was not timely or sufficient to support the necessary functions across the battlefield led to the development of a new paradigm for data handling. The core benefit of the paradigm shift is the ability to provide data to a shared space more timely and accurately than previously performed. Through a “smart pull” paradigm applications and users are given the control to pull the data they require without waiting on a push by the provider. By making the data available via the Task, Post in Parallel, Process in Parallel, and Use in Parallel (TPPU) paradigm, the process is enhanced. The diagram below illustrates the data flow and shared data process. For IWMDT the TPPU cycle will provide the intelligence data and integrated information much more rapidly allowing users to pull key data for model development and prediction assessments.

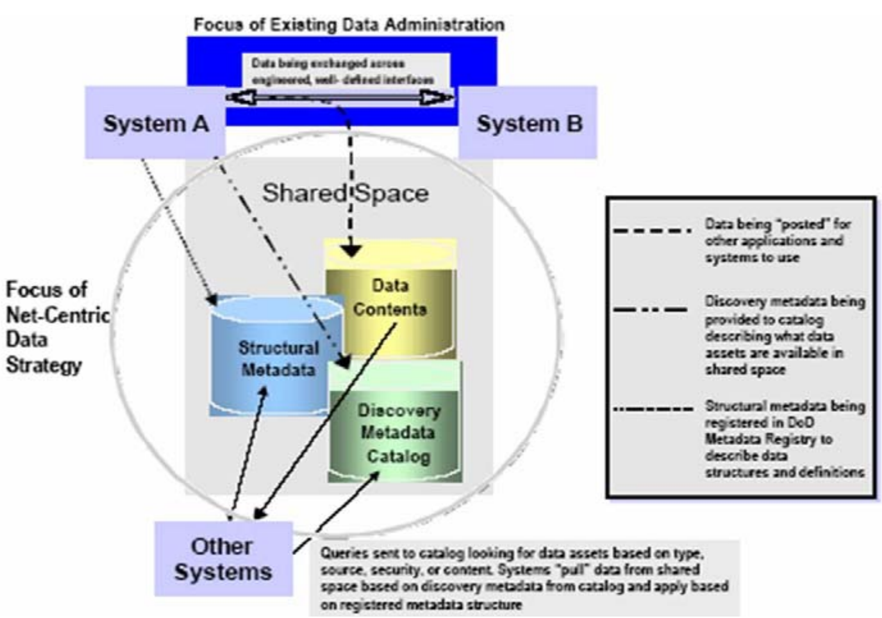


Figure 3. TPPU Environment (From HF Document, 2004.)

Previous processes required intelligence sources to process the data prior to posting it, and when posted it was usually stovepiped into areas that were not globally available. The vision of this new process is that the data will be in a shared space where components that adhere to the DoD standard interface documents request data and post data much more rapidly.

IWMDT has a goal of meeting the requirements of the Net-Centric Horizontal Fusion Portfolio[6] which are: make information available on a network that people depend on and trust, populate the network with new dynamic sources of information to defeat the enemy and deny the enemy advantages and exploit weaknesses.

F. WEB-SERVICES

The core of the web-services architecture is the ability of legacy applications to wrap modular software components in Internet communication protocols. This service enables legacy or non-web-enabled applications to run on a TCP/IP network, providing broader application use. Coupled with this simple goal are a number of other ancillary tasks such as automatic communication with other applications without human intervention, ability to deploy inside of firewalls, or run in a protected environment and the key is not to require a specific development tool or language. Figure 4 below illustrates the simple layout of the system.

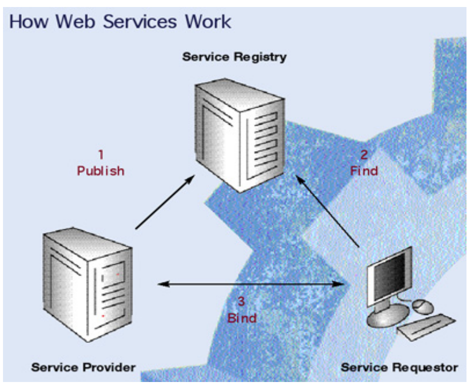


Figure 4. Web-Services Operation.

---

6 http://www.stsc.hill.af.mil/crosstalk/2004/01/0401Stenbit.pdf, May 2004

Key operations shown above are the magic behind web-services. The key operations are “publish”, “find” and “bind”. The first operation, “publish” represents the XML service description that the host or service provider creates making the information available to other modules on the web. Using Universal Description, Discovery and Integration (UDDI) protocols the next operation occurs, “find,” in this stage the service registry readies services descriptions for execution by other applications. Within the “find” operation UDDI protocols provide information on location, linking requirements and operational use. The last operation is the “bind & run” or “bind”. This operation is the result of the service requestor “finding” the information that was “published” and binding it to allow execution.

This activity is all possible because the data is meta-data tagged. The meta-data tagging process is shown below in Figure 5. Shown in the upper right corner is the benefit to the **Consumer**, which includes the automated search of data, focused format or content data, and the automated analysis of data content determination. The **Producer** is capable providing a wider variety of data formats and multi-media data, which is tagged and stored as a shared asset. In addition one of the strongest reasons to take advantage of the meta-data standards is to allow the **Developers** to build applications that post, process, display and exchange the tagged data. The use of meta-data is a benefit to all three core areas, the consumer, producer and developer.

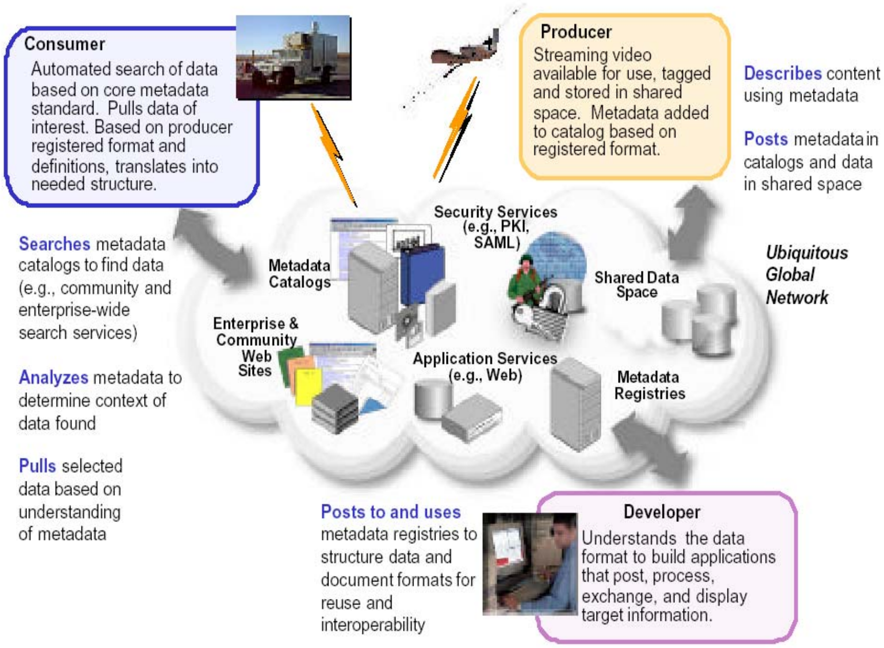


Figure 5. Meta-data process
(From CM Plan, 2003.)

G. SERVICE ORIENTED ARCHITECTURE

The last technology to discuss here is the Service Oriented Architecture Protocol (SOAP). This architecture provides services that promote loosely coupled interactions among software agents. The general rules for the entities are: a service is a well-defined self-contained unit of work, service consumer uses a service by making requests, a service provider provides services by processing requests, and a contract is a platform independent specification of the service. Primary characteristics of the services are: services are loosely coupled, services are location transparent, services are dynamically bound and discoverable, services are composable, and lastly services are platform independent.

To facilitate the necessary web-based services stated as goals of modern accessible applications and frameworks, many applications apply web services as an enabling technology. A primary reason that web service are selected for use in innovative designs is due to the platform independent, well defined, and self-contained unit of work methodology. The web services are implemented through the use of Web Services Description Language (WSDL) that provides standard ways of describing the contract between service consumer and provider. The publish-and-discover mechanism is provided by UDDI, and, the transmission responsibilities are provided through the use of SOAP. Shown below is a common SOAP architecture demonstrating this concept.

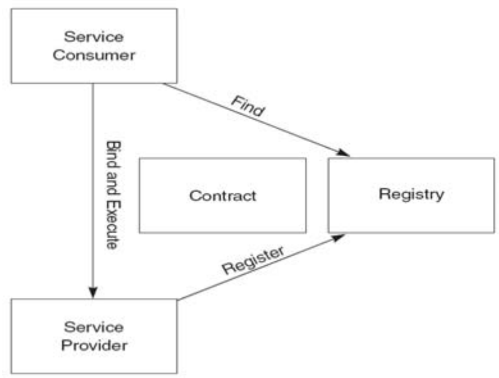


Figure 6. Service Oriented Architecture – IWMDT
(From IWMDT Program Review 03/15.)

THIS PAGE INTENTIONALLY LEFT BLANK

III. ANALYSIS OF REQUIREMENTS

A. LINGUISTIC DISCONTINUITY

"When you fail to plan, you plan to fail!" In software disciplines this typically means we fail to define the architecture and method in accordance with the stakeholders' requirements and the engineering clarity of the requirement process. The elicitation of requirements when correctly applied results in a more focused and less costly effort. But the assurance of this result is tied to many factors, some are controllable and many are not. For those areas that we can control and constrain, there must be a common understanding of the definitions.

Since the tower of Babel, man has spoken many languages, resulting in lost meaning and frustration between parties. Professor Richard Riehle[7] of Naval Postgraduate School (NPS) popularized a phrase among his students, "linguistic discontinuity" that addresses this phenomenon. He states:

> …. We begin a software effort by excavating the requirements, as seen by a client (or other engineer), in the language of that client. We translate that set of requirements, during analysis, into some other linguistic model. From there, we might impose yet other linguistic models based on input from other engineering disciplines. At some point, we create a design that attempts to make sense of the linguistically unique input up to this point. After several iterations, we think we have it right.
>
> Then we pass this on to a set of developers including programmers and integrators, who have yet another linguistic toolbox. This toolbox consists of programming languages, operating systems languages, database languages, etc. Sometimes there is more than one programming language involved. Whatever the programmer creates is eventually translated into an executable language that only the computer can read with ease. Yet the programmer, during the debugging process, must also be able to translate the executable language back to his/her own code set to make sense of errors. The testing team often has its own linguistic model….

[7] Richard Riehle, Naval Postgraduate, Visiting Professor, Computer Science

The message conveyed is, language is imperfectly exercised by imperfect people for imperfect reasons. But, straddled with this understanding we must learn to communicate effectively if not perfectly. The ability to capture requirements and respond to needs is limited by our understanding of the problem. Even if the user perfectly described his requirements the linguistic model employed by the engineers and other members of a development team would affect the results.

A linguistic model shapes each of us, and how we translate this model affects what we think and hear. In most cases the translation of phrases is not dissimilar enough to readily affect the understanding, but even small differences in the definition of requirements can have 2nd and 3rd order effects. An example of how linguistic models can have 2nd and 3rd order effects is demonstrated by examining the Ancient Greeks[8], and their unwillingness to use the number zero.

The Ancient Greek mathematicians are well known for their knowledge and learned contributions to every aspect of modern world education. But in the midst of this great body of knowledge was a limiting factor that prohibited these learned men from achieving even greater feats. The Greeks based their understanding of mathematics on geometry. This linguistic model bound them to a system that did not represent zero. Numbers were represented by geometric symbols and there was no shape for zero; or shapes for any number less than zero. Therefore, Greek mathematicians would not consider solving problems that are commonly solved today starting in middle school. These problems required the mathematician to evaluate zero or negative values and this was not possible in the Greek society.

Certainly we would not say that the Greeks were incapable of determining their requirements any more than we would tell a user that they are unable to determine their unique requirements. The Greeks ***chose*** not to have a zero, based on a gestalt philosophy that concluded that in math, as well as other aspects of life, one should not represent the existence of nothing. The Greeks thought that it is improper to represent "nothing" with something, if it is "nothing" it does not exist and

[8] http://saxakali.com/COLOR_ASP/discoverof.htm , 3 May 2004

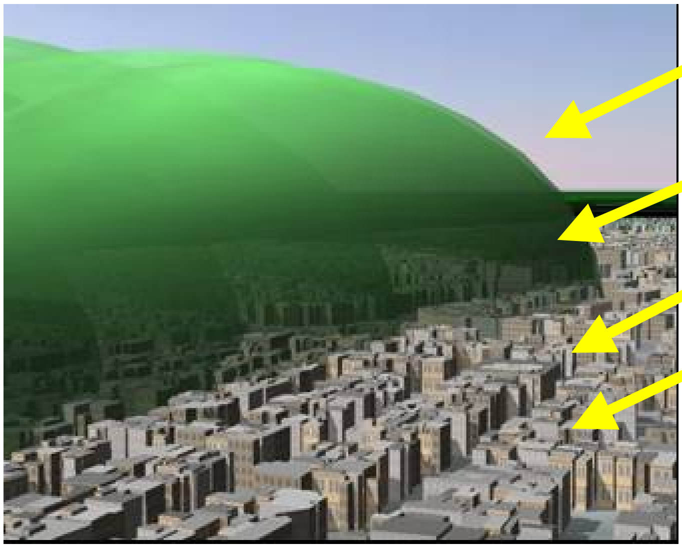
therefore is not necessary to account for it. Any other credo would invalidate their philosophies and violate their system of belief that did not account for the existence of nothing. Scholars of other faiths that allowed for the necessity of nothing, later formally introduced the concept of zero or "nothing" into the Greek worldview, and altered mathematics forever.

B. REQUIREMENT DEFINITION ANALYSIS

As Software Engineers, we must consider the impact of linguistic models and their impact on user's requirement statements. Because the user does not state a requirement that appears obvious to the engineer does not allow the engineer the license to modify the users' requirement without consent. Requirement definition is a difficult issue for engineers to broach with users and one that must be done with humility and an eagerness to serve the user's needs in spite of their perceived innate linguistic model limitations. With the "Greek" model in mind, we can appreciate that users still have a definite advantage over the engineers in the definition of their requirement.

A primary role of the Software Engineer is as a facilitator of discussions on what is possible given technology and data availability, not as a pipe for the user to pass requirements through. The Software Engineer must have the background in applicable disciplines to linguistically enhance the statements made by the user without losing the intent or increasing the risk or cost beyond the users' actual requirement.

Different users in the same geographic area will have different requirements for the prediction and assessment of WMD incidents as shown in Figure 7. The various users shown below each have unique requirements within the same geographic area. With an understanding of the varied environments, the Software Engineer can interpolate varied requirements that address each of the needs.



Protected	Unprotected
Aerial Entities	Aerial Entities
Mechanized Entities	Mechanized Entities
Dismounted Entities	Dismounted Entities
Non-Combatants	Non-Combatants

Figure 7. Myriad of User Requirements.

The user, the architect and the program team must share common understanding of the environments and the impacts required on the entities in the area. The effective communication of user requirements to the architect, and then the translation of the user requirements to the program team is critical. To understand the requirement in sufficient detail to develop code to support the incident, the requirement documents and subsequent documents must capture the nuances of the requirement as well as the stated requirements.

In this scenario users would be expected to convey the impact of agents on aerial units in a different manner than other units in the area. Agents move through the air in wind flow and as the heat and density of air changes the flow obviates some of the agent while massing other agent particles. Obviously the contamination at higher altitudes is more of an issue for the aerial units, but when they land the agents may impact ground units. This is where the user's requirements must be captured with clarity. The clarity depends on the shared understanding of the terms, concepts and requirements.

The author failed to accomplish this initially when developing the program goals and architectural designs for IWMDT. The definition and translation success was affected due to miscommunication and discontinuous goals between the users and the author. Though the author's intent was to clearly define a requirement for a single tool that would replicate all functionality of the existing HPAC tool, he translated a requirement for a web-services net-centric architecture to the development team. The result was much better than the intended goal because it used burgeoning technologies and enabled much broader application integration, but, the ineffectiveness of the translation was still proves the point of miscommunication and its impact on requirements.

A primary reason the author failed to clearly articulate his requirement was his bi-lingual familiarity of both the warfighters operational environment and the programmer's development environment. The author was too familiar with both environments and he failed to clarify the key terms, subsequently the development team assumed he understood the terms as they understood them. When the requirement documents were delivered to the author it was obvious that the goals were incongruent.

When the author considered the possible discontinuity of the parties and described the intended goals and requirements in non-domain-specific terms the problems were largely abated. Within the specific domain (developers, users, etc) terms are normally commonly understood and meaning is easily transferred between team members, the same is typically not true outside of the specific domain. In this example the typical user is a military warfighter or a first responder who is rarely an expert at WMD. Likewise the typical developer is a C++ or Java programmer in San Diego CA, with little or no military experience. And the author is representative of a typical government lead that is expected to have thorough knowledge of both domains. Though this is rarely true, it is an expectation of the other two pieces of the team and therefore should be accounted for in the process.

With the shared experience and understanding of both environments it is incumbent on the software engineer to ensure that the WME experience, military experience and software engineering knowledge are seamlessly integrated. The author was capable of transferring meaning between all three domains and maintaining delineated domain specific meaning for terms, but in this case he did not accomplish the task early enough. Because the government lead is the only person on the team that is expected to share a common understanding, they are a key to the effort. In most cases, the government lead is not a software engineer and is not prepared to accomplish this critical role. This is a subject for other papers to discuss the validity of the government lead nomination process. The miscommunication in this case actually benefited the project because it exposed the author to possibilities he had not considered. Most participants in the development process have one domain that they are an expert in and very limited knowledge of the other aspects of the project.

An example of the impact of linguistic discontinuity and its impact on shared understanding is demonstrated with three seemingly innocent words. The three words, “hazards”, “thresholds” and “agents” are used to demonstrate the concept. These are common terms in all three domains, but each has a differing definition for the common phrase. To participate in the project all three must share a common definition of the terms. Failure to share a definition causes misunderstandings and discontinuity of the general and specific applications of the terms.

When considering the phrases the WME expert thinks a contour of agent contamination for “hazards” and “thresholds,” and specific chemical or biological solutions for “agent”. The user thinks of minefields and no-go terrain for “hazard”, and personnel flow into the operational area for “threshold”. The software engineers has yet a third use of the terms, referring to malicious code and bandwidth issues as they consider hazards or thresholds, and to agent-based systems for “agent.” These differences can cause an incongruence of the user expectation and the developed product, more complex terms and concepts will be further exacerbated by the discontinuity.

Professor Riehle pointed out the requirement for embedded engineering disciplines within the software discipline. Within the software effort for IWMDT we have chemical, biological, nuclear, and structural disciplines. Consistent with Professor Riehle's comments and our software engineering community a respect for other disciplines and their respective experts is an important criterion for successful team-based projects.

Professor Riehle further states, "…software has become an equal partner in the design of modern systems of all kinds." The specific reference to software illustrates a maturity of the software discipline to the point that the software experts are viewed as important as the structural or major systems experts. Though this certainly is not a unique idea, the inference that linguistic discontinuity as we move across the spectrum of "equal partners" affects the process is a less researched and defined concept.

The two tables on the following page represent a hypothetical situation that could occur as we define the requirements for IWMDT with the users, software engineers and programmers. Though this is only an example of a typical requirement process it presents the basic premise of lost meaning as words are passed through domains.

User Stated Objective	Software Engineer Hears	Intended Meaning	Potential Impact
I want to be able to use a browser to run the application	User has a requirement for a browser-like GUI. Requirement is to develop a user-friendly browser-like GUI while maintaining the current client-server architecture.	I do not want to download or otherwise install any software on my computer	The user is forced to load over 2 GB of program data, and maps. Network administrator denies the user the right to load the software on his local machine because it requires ports be opened, his local machine does not have 2GB of user available space.
I need the ability to predict which way I should move my troops	Provide a course of action tool that tells the decision maker where and when to move his/her troops to minimize exposure	Don't give me the answer, give me the information for me to make a decision	If the tool is driving the movement of troops and equipment than the only consideration is the WME. This is not the way that we must operate on the battlefield. The WME is only one of many considerations for troop movement. Commander will not be hostage to any one tool.

Table 1. Linguistic discontinuity – Software Engineer.

User Stated Objective	Programmer Hears	Intended Meaning	Potential Impact
I want to be able to use a browser to run the application	I need to redesign my entire architecture to take the current thick client architecture and develop thin client architecture. The requirement is to develop a system that performs all actions except display on a server in a remote location, and post to the local client	I do not want to download or otherwise install any software on my computer	Though the programmer is closer to the meaning of the user, this is not necessarily what the user requested. By using a thin client architecture the programmer is eliminating the requirement to load or install any software, but he is introducing the bandwidth problem. If the user does not have network access, this system is of no value.
I need the ability to predict which way I should move my troops	Develop a tool that can be user-selected to layer plume assessments and route impact.	Don't give me the answer, give me the information for me to make a decision	The introduction of layers and user-selected assessments is a good method, but coupled with the route impact this becomes a very expensive venture. Potential for loss of focus and costly development is very high.

Table 2. Linguistic discontinuity – Programmer.

This is a simple example of how easily the linguistic meaning is continuously challenged as we worked through the specific requirements for IWMDT and represents a typical community phenomenon. The simple manner that causes the lost of meaning illustrates Professor Riehle's statements and coining of the phrase "linguistic discontinuity". But, the problem is more complicated than just the definition of a few words or even the concepts that users want to express. Assuming we work through the language differences and we define the specific users requirement we now have the larger issue of determining if the user understands their own linguistic bias.

Much of the poor requirement solicitation prevalent in the government process is due to the disparate requirements and the geographically unique environments. The requirement solicitation is normally an indirect process because of the evolutionary nature of the requirement maturity. Even though all modern military units account for and train to fight within a WME environment, the methods and training frequency is largely based on each commander's assessment. Each commander is responsible for their own unit's readiness, and determination of the potential threat for operating within any particular environment.

In the military we must consider the potential to operate within a contaminated environment every time we deploy. Just as true with other potential threats or impacts, commanders must plan their training and operational concepts to accommodate the effects of WME. This planning is normally insufficient for the actual effects of WME assuming the units were faced with actual affected area. A primary reason why the effect of WME is underestimated in training and planning is due to the low potential for occurrence. One of the roles of the military leader is to assess the potential for a given impairment and then train and plan to overcome the obstacle.

With a lower than average potential for occurrence commanders do not emphasize training that would unduly affect normal operating procedures, by training for the low potential WME environment. This is not an indictment on the commanders it is a statement of efficiency for limited training opportunities. If we accept the authors postulation that it is not limited vision that causes a lack of training than Higher Commands must confer a sense of urgency to the probability of WME in order to get it trained more frequently.

Assuming we are able to affect the frequency of the training, it now becomes even more critical that the data and information that results from the training be relevant. This data represents the entities and events that the commanders are already considering in other areas of the training and must be realistic to be believed. The accessibility and extensible of this data is a key in the integration of WME into other battlefield operations.

C. DATA REQUIREMENTS

Because DTRA is not an intelligence provider they rely heavily on other Agencies and systems to provide the key data required for IWMDT. This data is gathered through classified and unclassified means and is propagated across the world to support Major Commands and Key Allies. Because much of this data is available only within the Commands, effective application of IWMDT will rely on establishing processes that allow the IWMDT operator to gain access to this data. This data includes terrain, hard target graphics, agent information, weather data and any key information maintained by the Commands. In DoD a key overarching objective is to enable data suppliers to provide critical, timely, affordable, verified and validated data to the user community. Based on this objective the goal is by providing validated and verified data it will enable models and simulations use to gain credibility and to support more key decisions.

To accomplish this, the data and information providers must understand the standard that they are expected to meet when they store and distribute the data to the users. These standards address data element definitions, data dictionaries, data models etc. There is not a universal standard; therefore software development agencies such as DTRA are not guided to a standard method when developing their systems. It is imperative that developers make every attempt to adhere to commercial standards allowing them to benefit from the tremendous amount of data that is commonly produced outside of DoD. Hence, most software developers across DoD maintain a working knowledge of both the DoD policies as well as the commercial standards.

The DoD Master Plan for Models and Simulations defines data as: specification of facts, parameters, values, concepts, or instructions in a formalized manner suitable for communications, interpretation, or processing by humans or by automatic means. Closely related is “information”, which is defined as, data in context, related to a specific purpose. In lieu of a universal DoD standard, DTRA developed interfaces with common commercial data standards that are in broad use across the domain community. The decision to use commercial standards enabled the developers to expand their design extensibility to provide for future integration efforts. These standards are required to provide accurate representation of sea, air and land entities both internally as well as in external integration of other applications with IWMDT.

IWMDT is designed to allow the accurate prediction of hazard assessments across blue water, brown water and land, and to include three-dimensional objects. The data requirement for IWMDT requires authoritative representation of the permanent and semi-permanent man-made features as well as the natural terrain. The data must allow IWMDT to process model data for generating, moving, dispersing, and dissipating atmospheric phenomena.

1. Specific Data Standards

The IWMDT data storage and retrieval system is based on existing databases developed under a separate program entitled Integrated Target Planning Tool Set (ITPTS). This database structure supports a consolidated targeting system, using a tree-node database tree structure. The use of a tree structure provides separate root nodes for public folders and personal folders. Following the Windows method of management, some nodes contain no specific files but exist for logical groupings.

Though the database module does not benefit from the organization the database navigator does. Having designed it based on Windows structure methods: the structure is viewable as a hard drive directory with the root node appearing as a drive letter like other drive directories.

The database is organized in a hierarchy of nodes (rather than tables), which introduces the following characteristics: the primary nodes are designed in parent-child relationships, which reference other nodes by key linking. Within the nodes the databases contain node types (target, session, etc.), properties (string/value pair metadata about the node), comments (system and user-generated information), approvals (contents has been examined by an approving authority), and files (analysis data, multiple files per node, BLOB-format). The use of the node hierarchy provides us at least two key advantages, we retain analysis data in a common location (useful for centralized backup and replication), and we retain the classification pedigree for the data (useful for handling multi-level security of data and providing centralized access-control and managing "need to know").

The IWMDT server communicates with the database through a CORBA interface, running under a JBOSS application environment. As a JBOSS service it exposes its factory classes and naming context to other services across the network. Components across the IWMDT toolset have access to the information using the "NodeEntity" bean, which uses CORBA to interact with the persistent storage.

The database structure provides the ability to organize and retain the relationships between analysis projects, provide separate domains for users, groups (future), public analysis projects, and provide declarative security (future). The database also allows projects to be stored in their native format without dictating rules or data types. Through the use of meta-data the database is capable of allowing projects to be queried, and to interrogate other applications without understanding the native format.

2. Data Composibility

Even when all stated standards are adhered to, all data must be re-validated when introduced to new applications. As a premier Chemical, Biological and Nuclear Agency within DoD, DTRA has invested large amounts of money and time in the development and credibility of the source term data. IWMDT has ensured that this research and credibility has transferred into their tool by maintaining the same source definitions and data from previous bodies of work. This work has been validated in the field, laboratory and through many series of analysis and comparison in HPAC.

But in spite of the largess of DTRA to provide this data to other users and models, the data must be validated and verified each time it is applied to new applications. Because the data is not isomorphic, it is unable to aptly evolve to meet nuances that are introduced when we apply different criteria in various models. So, when IWMDT matures past the R&D project into a deployable application, the data will require recertification as if it were a new source of data. Of course the previous validity is of value and is not discarded, but the application of the data in this particular development environment will be tested.

Specific data that needs certified includes not only the standard weaponized agents but also all of the Industrial Toxic Chemicals (TIC). The TICs are not typically as dangerous as the other agents but with extended access they can be lethal. There are over fifty TICs included in the standard library and just as is the case with the chemical and biological agents, more can be added by creating material files. The creation of material files is not a novice chore and should not be done without expert supervision to ensure that the complete agent property value is correctly captured.

D. WEATHER REQUIREMENTS

Source and agent data is an important aspect of predicting WME hazards, but without a reliable wind model the agents are not properly transported or dispersed.

Within IWMDT the reliance on weather is an integral part of the all assessments. Three different instantiations of weather are applied depending on the separate requirements. The three types for weather forecasts are: historical weather (climatology), forecast weather (numerical weather predictions), and current weather (observations).

Time permitting the user should apply all three of these weather data sources to their solution as shown in Figure 8 on the following page. Each source is applied at a different stage in the model use. As shown on the next page, a suggested timeline process is applied to the final development of the prediction. The figure illustrates the applicability of weather sources depending on the time to execution. For planning purposes or to anticipate events within a five-day period, forecast weather is the optimal source. For a post-event or real-time response, the user should apply current observations.

The use of multi-layered time sequenced weather is a unique element of IWMDT (drawn in from HPAC) because hazard prediction and the weather correspond with the estimate of confidence. The weather model bases this confidence on estimates of the uncertainty inherent in the forecast weather data. The estimates are calculated using real time probabilistic methods (for example, ensembling techniques) or via empirical models embedded within the software. This is a unique element of HPAC model (as well as IWMDT which incorporates HPAC) because hazard prediction and the weather correspond with the estimate of confidence. HPAC and IWMDT base confidences on estimates of the uncertainty inherent in the forecast weather data. These estimates are calculated using real time probabilistic methods (for example, ensembling techniques) or via empirical models embedded within the software.

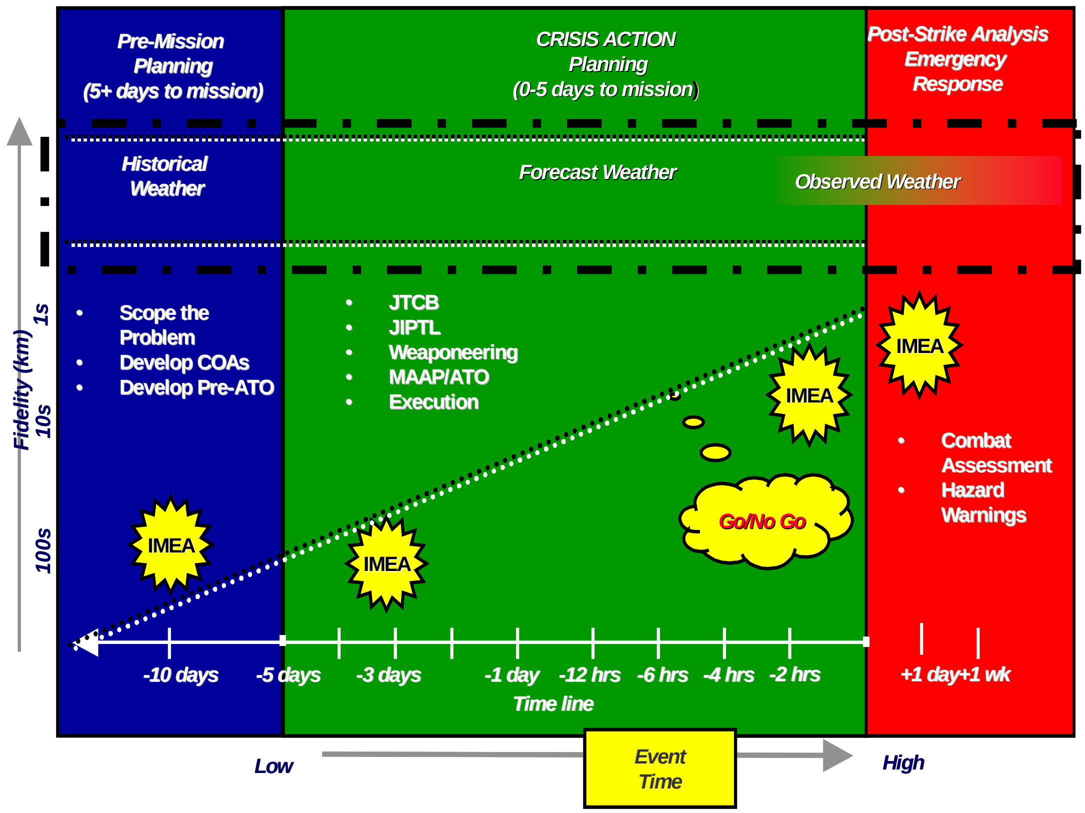


Figure 8. HPAC Weather Sources Use (From IMEA Tng Pkg 2001.)

There are five major providers of the weather data that IWMDT utilizes for assessments. These are:

1. Navy Operational Global Atmospheric Prediction System (NOGAPS) – Global coverage, 80-kilometer resolution
2. Navy model, Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) – Covers most areas of interest, 9-27-kilometer resolution, Navy model
3. Mesoscale Model Version 5 (MM5) – Covers most areas of interest, 5-45-kilometer resolution, Air Force model
4. Regional Atmospheric Modeling System (RAMS) - Covers small areas of interest, 1-10 km resolution
5. Operational Multi-scale Environment Model with Grid Adaptivity (OMEGA) – Covers small areas of interest, 1-10 km resolution. DTRA meteorologists generate RAMS or OMEGA products upon special request by users.

E. MAPPING REQUIREMENTS

The previous sections discussed the importance of the agent data and weather data, even with this well defined and populated, if we cannot present the information in a consistent and user-friendly manner the value is lost. One of the key goals of IWMDT was to develop the application around a central mapping system. The choice of standards was to use ARCGIS from ESRI. According to the ESRI ArcGIS website, ARCGIS is “a scalable system of software for geographic data for every organization-from an individual to a globally distributed network of people”. This common map interface must be functional for targeting, which involves scaling across a broad range of map scales as well as providing high-resolution imagery resolution.

In a recently released document outlining the framework for IWMDT, the following map requirements were elicited as shown in Table 3.

Provide ability to add (V0.5)/move /edit /remove IWMDT features like incidents on map	Provide ability to Pan, Scroll, Zoom in, Zoom out, and Zoom to full extent, Zoom to layer, Zoom to scale, Center on point, Zoom previous, and Refresh map (V0.5)
Display CADRG/CIB and other 3rd party standard GIS formats on map (V0.5)	Provide ability to set display options for IWMDT features like symbol size and symbol types
Provide ability to display a map (V0.5)	Provide ability to export map and legend images to user's computer (V0.5)
Provide ability to hide and show layers on map (V0.5)	Provide ability to add/remove/reorder layers on map (V0.5)
Use 0.5 mapping services with minimum porting to ESRI 9/CJMTK	Manage new mapping requirements to support new subsystems GIS capabilities
Provide ability to add points (V0.5), lines, polygon, shapefiles (V0.5), and raster data as layer on map	Provide ability to set transparency for each mapping service (V0.5)
Provide ability to export layers as ArcIMS Image (V0.5) and Feature services Provide ability to display multiple mapping services at a time (V0.5)	Provide ability to hide and show each mapping service (V0.5) Provide mapping capabilities as a web service (V0.5)
Provide ability to add/remove/reorder links to mapping services (V0.5)	Interface with JIVA-V

Table 3 Maintained Map GUI Features (From IWMDT CM Plan.)

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DEVELOPMENT OF THE ARCHITECTURE

A. REQUIREMENTS BASED PROTOTYPING

Typically in requirements engineering, prototyping is used to involve the user in the development process in an iterative process. By developing a user interface the process allows users to evaluate the effectiveness of the proposed system early in the development process making it is easier to modify the design. Correctly applied evolutionary prototyping should gather a correct and consistent set of requirements. The process tends to strengthen the design process by clarifying the stated requirements and identifying the approach to meeting those requirements. But, a drawback to this approach is the requirement creep that typically occurs when gathering requirements and other requirements are spawned.

One of the ways to counter-balance the impact of this dilemma is through the use of domain analysis. For the IWMDT effort, much of this was simplified by the fact that the author was the architect, user and decision-maker. As discussed earlier in the thesis, the author had two complementary roles, one role was to go forward to Commands and use the tools that he developed under the first role of COTR.

Not considering a unique situation as we had in IWMDT, the domain analysis is essential in designing high-quality software tools. The intent of domain analysis is to help the development team understand the requirements, identify the fundamental abstractions, verify the design and direct the implementation. Implementation of this task requires issues such as linguistic discontinuity to be minimized through recognition and active response.

In an ironic twist of literary application, it is humorous to note that even the term “domain” is not consistently defined within the software community. As we attempt to clarify terms, it is ironic that even the concept we are employing can be ambiguous. Depending on the view we are considering, the term morphs into various derivatives of the same concept. The key is to recognize whose view we are considering. From a user-based view, the term domain analysis refers to understanding the background of the end-user. Typically a clear way to articulate this is through use cases. The use case allows the actors, events and responses to be graphically portrayed, with their relationships demonstrated. From the perspective of the domain engineering community, domain analysis refers to studying the background knowledge necessary to solve the software design problem. And yet in a third perspective the term may refer to the evaluation of available hardware/software technology. Just as pointed out in other areas of the thesis, clarity of terms and requirements is critically important in the process of development.

Just as we need a short context explanation to agree on the definition of the term “domain”, we also need to understand the context for the term “architecture”. In 1969, Fred Brooks and Ken Iverson[9] described the term as "conceptual structure of a computer...as seen by the programmer". Less than two years later Brooks offered a more defined definition for architecture as, "the complete and detailed specification of the user interface. For a computer, this is the programming manual, for a compiler, it is the language manual... for the entire system it is the union of the manuals the user must consult to do his entire job." A contemporary of this gentleman, Gene Blaauw[10] offered a clear distinction between architecture and implementation. Quoting Blaauw, Brooks writes, "Where architecture tells what happens, implementation tells how it is made to happen." Because of the object-oriented nature of IWMDT this is a very important distinction. Because the evolution of the IWMDT capability remains focused on architecture, it is obviously important to ensure we understand the architecture.

The design view consists of three primary capability-based functional areas. The three areas are consequence assessment (CA), targeting support (TS) and nuclear phenomenology. Each of these areas is based on stand-alone capabilities that has been thoroughly tested and VVA for use worldwide. The architecture areas

---

[9] http://www.sei.cmu.edu/architecture/roots.html, May 2004

[10] http://www.research.ibm.com/journal/rd/441/amdahl.pdf, May 2004

are demonstrated in the diagram on the next page. The CA area migrated much of the HPAC capabilities, the TS migrated much of the IMEA capabilities, and the nuclear area migrated Integrated Nuclear Capability Assessment (INCA) and nuclear modules out of HPAC. Though the challenge of addressing the need for a WME assessment tool on-site was first addressed in 1991 after the Gulf War[11], IWMDT addresses the new innovative needs for providing shared databases and reducing the technology burden. The new implementation has only two user requirements: user must have network access, and second, user must have a standard windows compatible browser. This is simple, direct and easily understood, but bearing in mind our linguistic discontinuity is it really this simple?

The requirement is simply stated and yet broad enough to allow the engineers to perform software engineering magic. The magic is taking stand-alone tools each written against different requirements, with different definitions of common terms, without a common mapping protocol and integrating the tools into a common web-services based tool. This tool is expected to meet current capabilities of each stand-alone tool within a common mapping framework and a common GUI.

These requirements alone make this a daunting task, but consider that the project requirements are mostly pre-determined by the existing capabilities of the stand-alone tools integrated in the tool, and this becomes a real challenge. The key in this project must be less focused on previous validated user requirements, which are largely pre-determined for IWMDT, and more on the engineering approach to integrate the tools and provide a common front end.

Presented in Figure 9 is the IWMDT design architecture. Following the figure is the design methodology, which is presented as a short description of the three major functional area tools. The description of HPAC is more detailed as the HPAC architecture and program stream was the baseline methodology used in IWMDT.

---

[11] Toth, J.A. (1999). Hazard Prediction and Assessment Capability (HPAC)

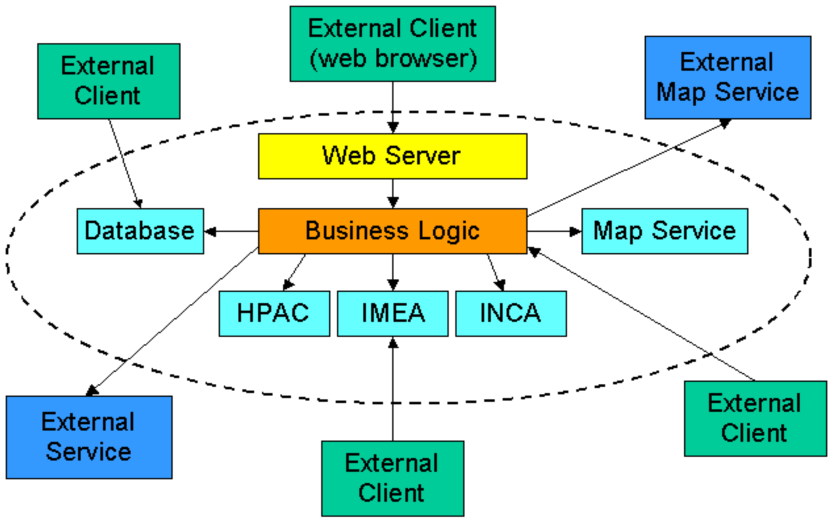


Figure 9. IWMDT Model Integration Diagram.
(From IWMDT How To Manual, 2003.)

B. HPAC ARCHITECTURE DEVELOPMENT

To this point in the thesis, we have spoken of the post Gulf War technologies and looked at the requirements for new tools and transformation of existing tools. We then looked at some of the shortcomings of the current system as defined by the author's experiences. After deploying on operational support mission representing DTRA for over two years, it is obvious to the author that in spite of having quality tools, the inability to locally host or remotely access this data was an impediment to DTRA success. This architecture was designed to consist of three areas of interest. We now look at the architectural methods and concepts that define the tools that were integrated into the IWMDT architecture.

The first tool we will look at is the HPAC tool. It is important to understand the design of HPAC because many of the methods and processes are common in IWMDT and HPAC. The original architect stated three guiding goals for the architecture, portability, extensibility and flexibility. The descriptions below describe the intent of the goals and the process of adherence.

1. Portability

The myriad of requirements from a rapidly growing number of users in the 1990s presented the early engineers with daunting challenges for portability. One of the immature attempts included attempting to use a cross platform development environment, MainWin from MainSoft[12], to build a UNIX version. Due to the required time to accomplish this effort the attempt was abandoned. Developers soon lagged behind the Windows implementation and achieved incongruent results from the differences in FORTRAN compilers and the way dynamic link libraries (DLLs) were handled. The engineers also concluded that the resulting application was an unpleasant UNIX user experience due to significant behavioral differences. A follow-on effort to port only the calculation engine to UNIX was more successful but also was abandoned due to performance lag in spite of the increased calculation ability on UNIX hardware[13]. Using emulation environments allowed execution of the client Win32 application under Unix Operating System, and eliminated the expected performance degradation of calculations under an emulated environment.

2. Extensibility

Design documents in 1991 identified the requirement to allow new plug-in source models without requiring system rebuilds. A core module of HPAC is the Second-order Closure Integrated Puff (SCIPUFF), a Lagrangian puff dispersion model developed by Titan System Corporation[14]. This module requires detailed data of meteorology, terrain, material files as well as other inputs be fed to SCIPUFF to calculate the transport and dispersion. From this calculation the user is provided tracks, material concentrations, depositions, and doses. As new source models are improved they are added to the model to allow improved analysis, without an extensible architecture, this would not be possible with code rebuilds.

---

[12] http://www.mainsoft.com/products/pdfs/datasheet.pdf

[13] http://wigner.cped.ornl.gov/HPAC/info/paper.tm.pdf

[14] SCIPUFF Dispersion Model, http://www.titan.com

3. Flexibility

Original documents indicate that HPAC system requirements elicited three basic modes of deployment; a standalone system with no network connection, a client-server arrangement with data available on the client but calculations performed on a server (heavy client), and a client-server environment with data downloaded to the client from the server and calculations performed on the server (thin client). All three of these methods are viable deployment options for IWMDT, though much of the web-services methodology is lost on all but the last method. An enabler for allowing HPAC to demonstrate the three goals mentioned here is the use of CORBA services as shown below. This architecture discussed later in the chapter is ideal for the services and methods that HPAC required.

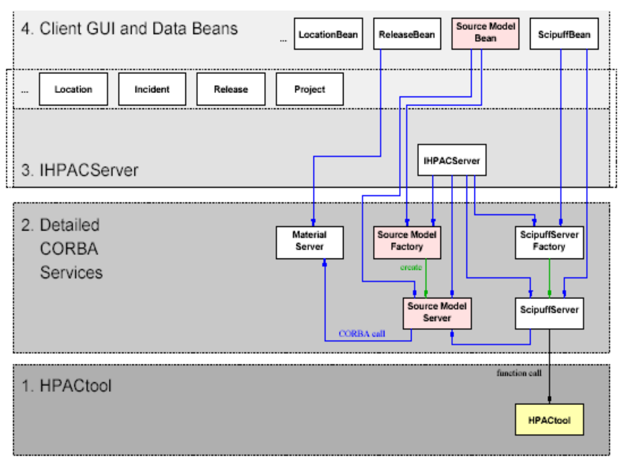


Figure 10. HPAC Architecture Activities (From Pgm Plan, 2003.)

C. HPAC-WARFIGHTER ARCHITECTURE DEVELOPMENT

In 2002, the next significant architecture generation leading to IWMDT was introduced. The 2002 development of HPAC-Warfighter[15] was introduced as the first web-browser based HPAC tool Figure 11. In collaboration with key SAIC developers, the author developed the architecture and operational vision for allowing web-based access to HPAC functionality. The author's vision was based on his previous

15 Developed as an intermediate process under the Battlefield Casualty Federate (BCF) effort. SAIC developed under BCF program at request of Major Ric Jones, COTR

experiences of deploying forward and being unable to load the required software on a locally controlled network. An issue that lead-up exercise to the Feb 2003 deployment to Iraqi Freedom identified as a requirement for forward deployed individuals.

As the senior WME hazard assessment analyst in the combat theater the author used the distributed HPAC-Warfighter tool in Iraqi Freedom 2003. The architecture did not meet the author's needs completely but the tool served as a very useful precursor to the development of IWMDT requirements and expectations.

Understanding the security implications of accessing networks and the sensitivity toward loading software on local workstations, the author's design document and requirement statement was specifically focused on web-services. The focus was the elimination of local load requirements, replaced with shared, accessible, visible data and processes. The early effort was far more focused and limited in the application of web services than the eventual development of IWMDT, but served as a vital impetus to the program. This was the most prominent of the follow-on efforts but shared development between a series of efforts made this possible.

The HPAC-Warfighter effort contributed to a significant follow-on effort entitled Battlefield Casualty Federate (BCF). Based on the incredible innovation and ingenuity of key software engineers and managers from Science Applications International Corporation (SAIC)[16], and the author's vision, DTRA was able to continue to expand the tool in manner that later would serve IWMDT members. HPAC-J introduced minor but significant changes to the HPAC tool. The changes provided a simple "unit" GUI button and associated ability to publish and subscribe unit information. The associated algorithms and logic to populate the data tables was previously incorporated in the code as a method for engineers to validate the code functionality but not associated with a GUI button.

---

[16] Continued HPAC-J effort - Mike Chagnon, Dave Walters, Mark Quan - SAIC

This effort bridged the technology from HPAC-J to the current IWMDT effort. The yellow button shown below in Figure 11 demonstrates the improved HPAC-J GUI functionality. This change allowed users to place units in their areas of concern and receive a tabulated or visual result of contamination. When linked to an active common operating picture (COP) this tool was capable of doing real-time unit specific plot assessments.

The results from these efforts were validated during Ulchi Focus Lens (UFL) 02, United States Forces Korea (USFK) in a collaborative effort with the Defense Modeling Simulation Office (DMSO)[17], but due to local network constraints further work effort was halted.

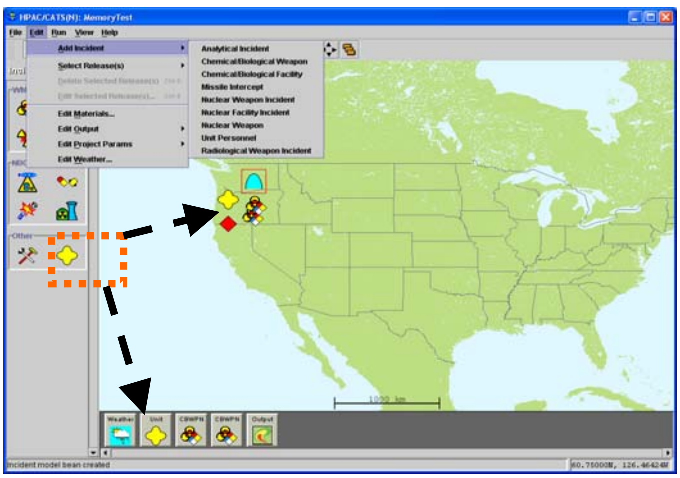


Figure 11. HPAC-J GUI ([From Prgm Review – SAIC, 2003].)

The knowledge gained during these tests directly contributed to the current IWMDT effort. With the units location precisely determined the predicted effect of units or specific areas is more qualitative. The intent for this module enhancement was to allow users to receive GCCS tracks and automatically update icons and perform assessments of hazardous exposures. But without a direct feed from a GCCS-K system, the user must hand place or manually enter data for the units or sensors prior to assessing the hazardous exposure predictions.

[17] Collaborative DMSO/(MITRE) – DTRA effort 2002, USFK was test site UFL2002

Though this change had great utility and was implemented, the eventual goal of accessible, shared data was not yet achieved. With a taste for collaboration now, the author continued to push the envelope with the excellent SAIC development team. As the author's learning of software engineering and methods accelerated in the distant learning program at NPS, his ability to ask the right questions improved. The improvement in his questions coupled with the understanding of the development team needs led to the refinement of IWMDT requirements.

D. IMEA ARCHITECTURE DEVELOPMENT

Just as the CBRNE community had requirements for the hazard prediction tool addressed by HPAC, the targeting community has requirements for predicting and controlling collateral effects and prompt response addressed by IMEA. Under the Counterproliferation (CP) ACTD, the principle sponsor U.S. European Command (USEUCOM) with support from U.S. Strategic Command (USSTRATCOM), tasked DTRA as Executive Agent to develop, integrate, demonstrate, and transition a suitable tool to the warfighters.

As the Executive Agent, DTRA contracted Applied Research Associates (ARA)[18] as the prime application developers. Other development organizations involved selected Department of Energy Laboratories, the Defense Advanced Research Projects Agency and the Defense Airborne Reconnaissance Office.

Based on demonstrated success on related projects, ARA chose the spiral development method for developing the Munitions Effectiveness Assessment (MEA) tool. The diagram below illustrates the five critical elements considered in the spiral development process. The first element is Requirements; this step is the basis for the other steps and relies on an accurate definition of the user requirements. The next step is Analysis; the effective translation by the engineer of the requirements into a top-level design is a key here. A successful translation of the requirements enables a robust development of component level designs. To this point the

---

18 http://www.ara.com/ May, 2004

engineer has developed the logic and requirement based templates, now the engineer must develop the code that supports the previous work. This stage is the Implementation stage. As the code is developed, the last step, Testing, is introduced. This is an iterative process testing at all levels of the program from unit through integration and eventually system level testing.

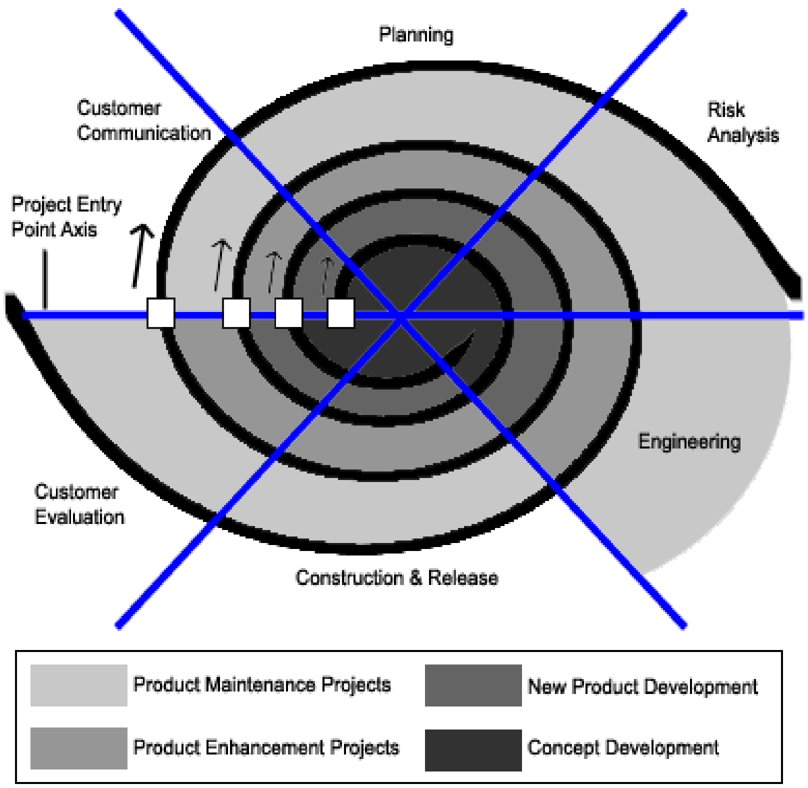


Figure 12. Spiral Development Process (ARA PGM Plan, 2003.)

In 1996, Phase I of the ACTD directed DTRA to employ current or very-near-term technology products in weapons, sensors, target planning, and collateral effects minimization. Additionally they were tasked to operationally demonstrate the capabilities against simulated biological agents housed in a soft, aboveground structure. The development provided significant engineering model and visualization capabilities for building/bunkers and tunnel modules.

As shown below the initial testing scenario evaluated the ability of the software tool to provide timely and accurate target planning support. The diagram below represents a typical attack scenario attacking a cut & cover bunker and a semi-hardened building.

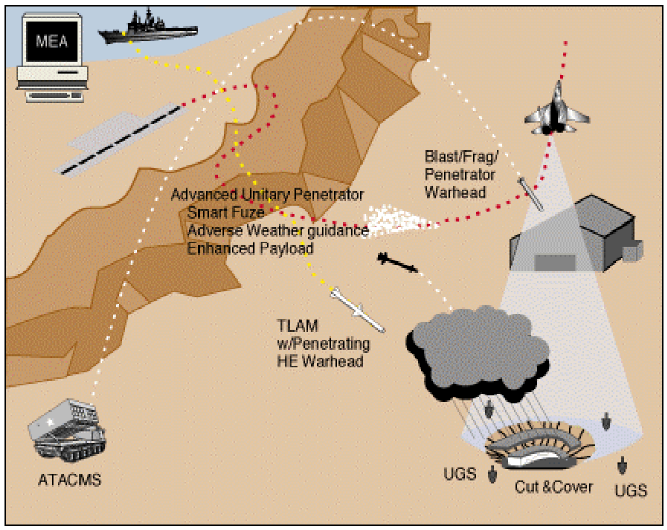


Figure 13. ACTD CONOPS for Phase I.

The demonstration was successfully completed and the enhanced models improved the accuracy and fidelity of the existing damage prediction algorithms. The testing also allowed extension of the current capability including additional weapon effects, weapon types, and damage mechanisms. Shortly after the Phase I testing, Phase II requirements were refined and schedules were established for enhancement of the software to meet additional challenging needs.

Phase II included sensors and data fusion for target planning and Battle Damage Assessment (BDA). Specifically the requirement addressed methodologies to assess weapon effectiveness for determining structural and functional damage/kills and agent dispersal/collateral effects. Additionally meteorological effects, enhancements in adverse weather weapon delivery, improved weapon penetration, warhead lethality, and fuzing was addressed. Figure 14 illustrates the notional concept of operations for Phase II.

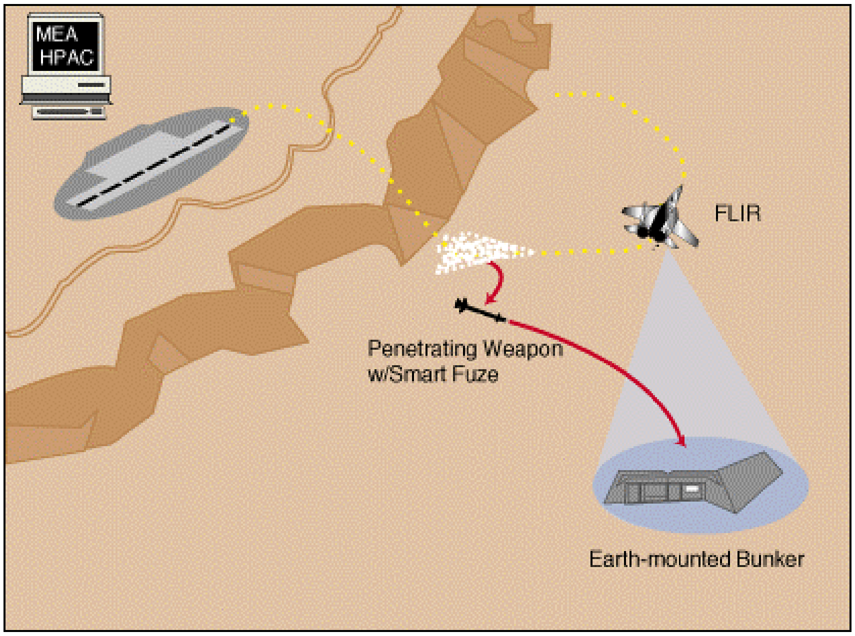


Figure 14. ACTD CONOPS for Phase II.

After the CP II ACTD was completed, DTRA continued to enhance the tool and in 2000 MEA was integrated with HPAC to become Integrated Munitions Effectiveness Assessment tool (IMEA). This integrated capability was carried forward to the IWMDT development effort.

E. IWMDT ARCHITECTURE DEVELOPMENT

Now with an understanding of the primary models that IWMDT consists of, we will look specifically at the IWMDT Architecture. The proposed design consists of four tiers. One of the primary formal goals of the effort according to DTRA documents is, "to make shared components available as platform independent components so that clients can connect to them through a standard interface."[19]

The informal architecture goals are to eliminate all local plug-in download requirements, eliminate any third-party costs to the user, implement web-services, and provide a common map-based environment for WME assessment.

[19] M&S Master Plan 2003 - DTRA

One of the early design decisions for the architecture was the choice to use CORBA[20] as the distributed object architecture. Other architectures were considered and eliminated for various reasons. DCOM was rejected because it is not generally platform independent, being primarily Windows operating system friendly. Java RMI[21] was rejected because it requires java code on client and server side, and a goal was not to require any code on the client side. The use of EJB was rejected due to the high server side resources, though it does scale well it was envisioned as being too high end for this application. EJB may be incorporated or employed in the future depending on the growth of the toolset. The use of CORBA is generally thought to be a vendor, platform, and language neutral choice and is consistent with the intent of this program. Web-services will certainly invalidate much of the CORBA activity in the near future but as of the development of this architecture, CORBA was still the better choice.

Based on a tiered approach the architecture for IWMDT invites a myriad of challenges. At each level of the architecture the integration and interoperability is exasperated by the dissimilar code structures of the component modules. Each component was designed and developed as a stand-alone tool, and each is a very complex and interdependent set of routines. Though the tools were developed as stand-alone applications, the interdependencies are well documented which does enable accurate interface documentation.

The IWMDT architecture has four tiers; Client, Web, Business, and Enterprise Information Systems (EIS) as shown in Figure 15. The Client Tier consists primarily of HTML and is the data presentation level. The Web Tier is primarily JSP pages and servlets designed to process the get and post requests. The Business Tier handles the core logic and interfaces with the data sources and various engineering models. The final tier, EIS performs the data storage and retrieval services from the calculation engines within the engineering models. In addition to the use of tiers as

[20] The Common Object Request Broker: Architecture and Specification, Revision 2.3.1, Object Management Group, Inc., October 1999.

[21] Java™ RMI Over IIOP, http://java.sun.com/products/rmi-iiop/index.html.

discussed above, IWMDT is broadly based on the use of a service-oriented architecture.

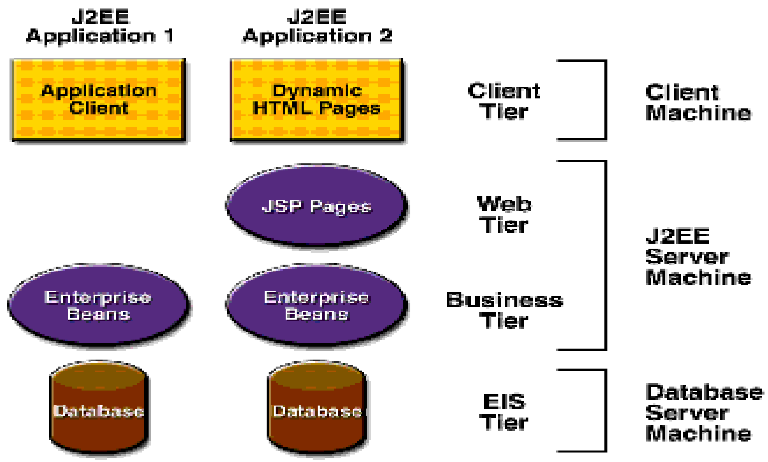


Figure 15. IWMDT Prototype Architecture. (From IWMDT Pgm Plan, 2003.)

1. Client Tier

The Client Tier provides the network API for applications, which allows applications to be written in a variety of languages. Within this tier numerous capabilities are addressed to include providing a consistent data standard, providing access for clients, providing directory services for toolbox components and providing a common look/feel.

2. Web Tier

Applications should keep data, control, and presentation logic as separate entities. The Model-View-Control pattern[22], is an example of this. shown in Figure 16. In the MVC pattern, a web server receives a GET or POST from the browser. The request is packaged and sent to a controller servlet that performs an action against the data and generates the result, which is rendered into HTML by a JSP viewer.

---

[22] http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html Apr, 2004

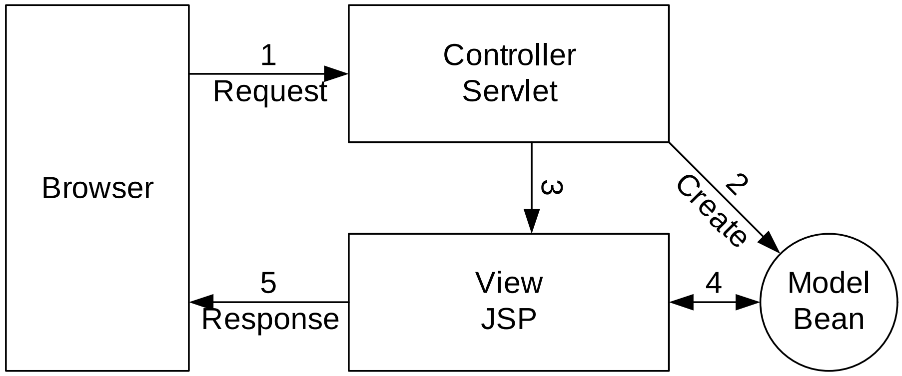


Figure 16. Model-View-Controller pattern.
(From IWMDT Dev TIM, 2003.)

3. Business Tier

Under the Business Tier there are two distinct component types, system components and low-level components. The system components are responsible for integrating low level components, providing high level interface for use by applications, implementing Web Service and CORBA interfaces, and causing the application to be extensible for other types of interfaces (Jini or RMI etc). Examples of direct implications of the capabilities provided in this tier are IHPAC and Weather services.

The second type of component that is employed is the low-level component type. The components within this level have a single focused responsibility. This focused approach allows each component to contain a primary algorithm for calculating responsibility. These types are primarily for internal IWMDT used by systems components, they provide a CORBA interface and if required enable an optional Web-Services interface. Examples of this would be SCIPUFF Server, and PDCALC.

4. EIS Tier (Subtier- Legacy Tier)

The next tier level is the EIS Tier consisting of tow sub-tiers. The first sub-tier is the Legacy Tier, which provides an application framework. This tier is responsible for providing a toolbox of GUI components as well as allowing rapid application development by supplying commonly used GUI components. Through this level we are provided with a set of interface data translators that allows for a common data communication mechanism. This level also provides an interface for data translators allowing data to be shared among components when their native data standards aren't compatible. Due to the open nature of the development design applications can have a GUI (browser or heavy client GIS) or be batch/script-based. Examples of this level are HPAC, CATS, IMEA, and WALTS

5. EIS Tier (Subtier - Utility Tier)

The second sub-tier is the Utility Tier, which allows distribution of data, and allows multiple users or servers to access data simultaneously. The data resides in this tier that contains the data used and generated by the Toolbox. This data normally physical resides in an external database such as Oracle or SQL Server or it may reside in flat files. An important aspect of the data storage is that it can reside at disparate locations, can be accessed via the Web or LAN, and it allows for new types of data to easily be integrated into the Toolbox. Examples of this level are: Weather, Deposition/Dosage, Population, and Terrain.

F. GRAPHICAL USER INTERFACE (GUI)

The tiers provide internal functionality, but the functionality of the interface is equally important. The graphical user interface (GUI) is very important to users, and may decide if they are willing to use the tool. The map format is ARC-GIS based on ESRI products loaded on the server side that do not require local download or plug-ins for the client. The main IWMDT interface is a simple map-based interface allowing layer tailoring on parallel frames while maintaining the map display in the central frame, as shown in Figure 17.

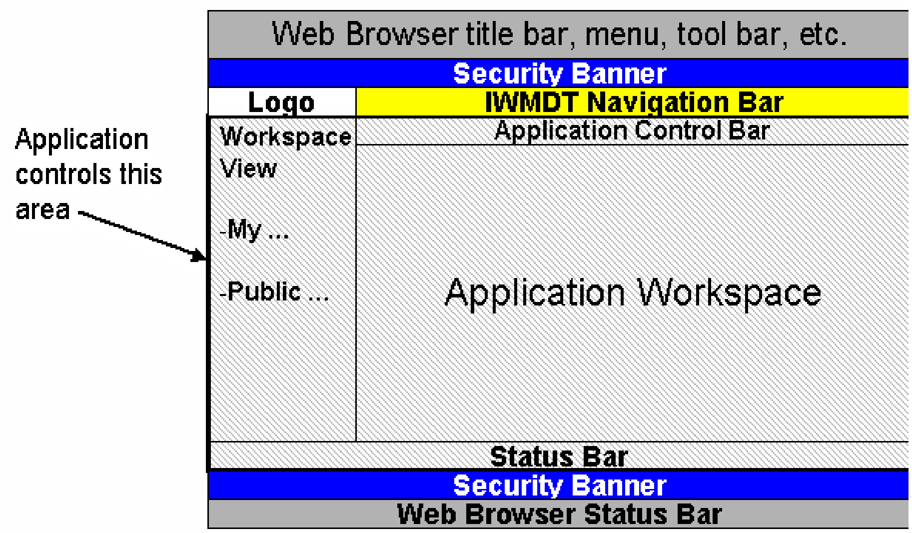


Figure 17. IWMDT Prototype GUI.
(From IWMDT Design Plan, 2003)

Though there is not a consistent preference for all users, for the sake of screen efficiency and based on expert opinion the best way to provide the information and allow tailored appearance was the use of frames. The effectiveness of GUIs for applications using Web-services is an area of research that continues to receive a great of attention across our field.[23]

Within the left frame is a set of boxes that is easily selected to tailor the map display for each user's requirements based on the availability of information at that level. As we change scale of the map, alternative scales are offered. The scales currently span from 1m data up to standard 1:250K map displays. The architecture allows users to reference imagery, maps or feature data. When performing analysis and plotting predicted contours it is very important to maintain geographic context. A prediction of 30KM in a northwest direction obviously is more relevant if we are plotting over Baghdad than if we are outside Bayji in central Iraq. In Figure 18 we see a number of other positive user-friendly features of this GUI.

The Main Window Layout consists of classification panes at top and bottom of window, application header pane below top classification pane, and the main application pane and status pane. The Sub-dialog Layout consists of classification

[23] V. Maciejewski and J. Zukowski, .Developing Rich User Interfaces for Software Services., Spidertop Technical Report TR101, Spidertop, Inc., October 2001.

panes at top and bottom of window, and navigation (summary) pane and working (detail) pane. The layers shown on the right side allow the user to easily identify the contours and isolate layers as desired (not shown). The "map-quest" appearance of the GUI allows an easily understood method of scaling and manipulation. Most users are familiar with the magnifying glass to expand or reduce the area of the map which is used in this design.

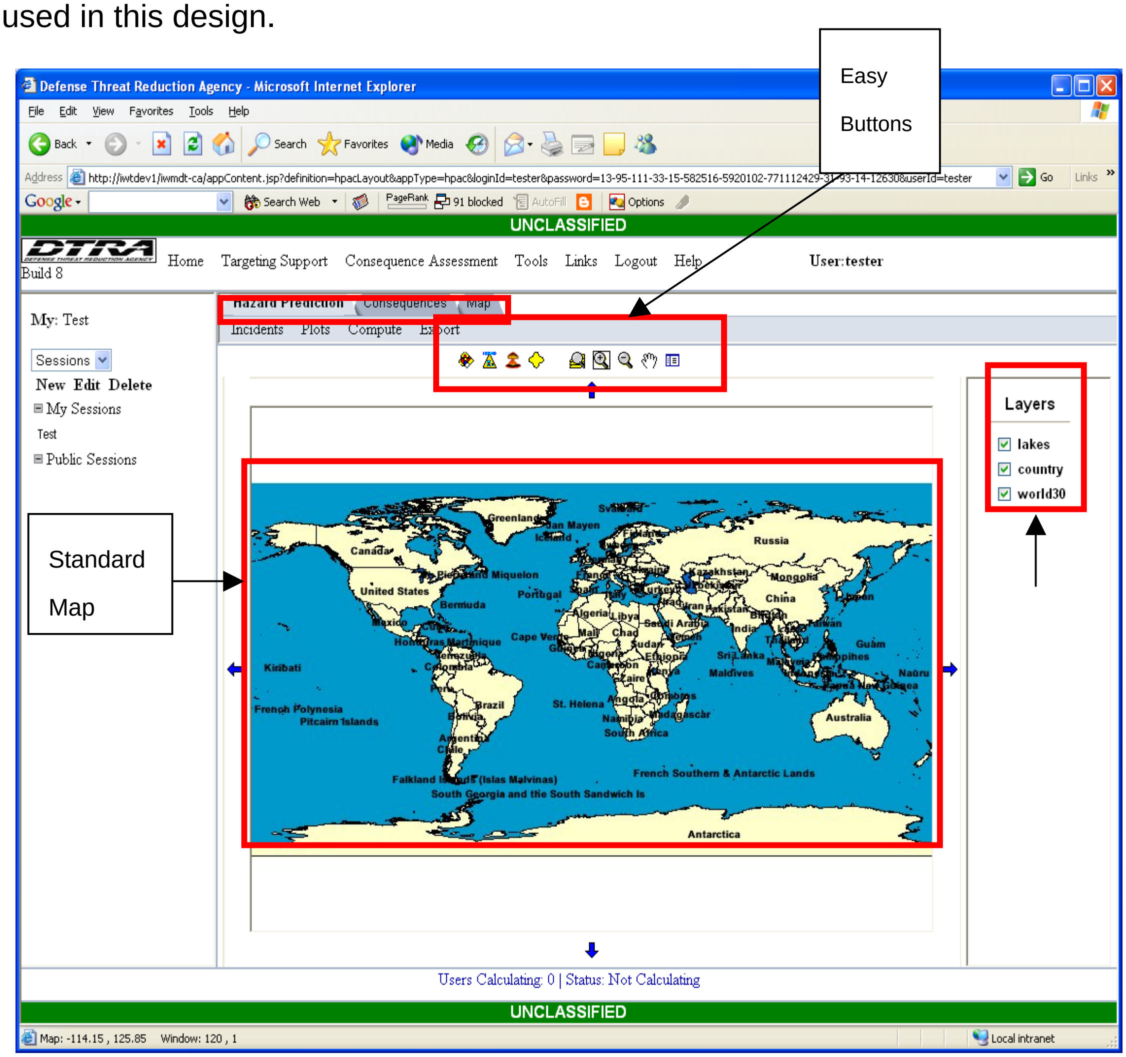


Figure 18. IWMDT GUI.
(Actual Screenshot)

G. ANALYSIS OF DESIGN DIAGRAMS

The current development effort is based on client-server implementation of supporting computational tools allowing access to the capabilities of separate tools in an integrated toolset using web services. At the conceptual level there are defined frameworks described by LISI[24] which distinguish the following layers: Isolated Systems (No physical connection exists), Connected Systems (Homogeneous product exchange is possible), Distributed Systems (Heterogeneous product exchange is possible), Integrated Systems (Shared applications and shared data), and Universal systems (Enterprise wide shared systems). The IWMDT design is similar to previous internal code design with a new web services GUI. The new GUI and the enterprise wide openness certainly provide justification to indicate that the system type is approaching “universal system”. This implies that we have enterprise wide sharing of data and processes. As shown in this document, it is certainly the goal of the IWMDT to accomplish this functionality. The task is not completed yet, but with continued excellence they are rapidly approaching the goal. In Figure 19 we see the top-level breadth of the IWMDT capability goals.

Figure 19. Integration of DTRA Tools into the GIG (IWMDT SDR)

---

24 http://www.DoDccrp.org/events/1999/1999CCRTS/pdf_files/track_5/049clark.pdf

During a recent Simulation International Standards Organization (SISO) panel discussion on priorities for M&S standards [25], Dr. Zeigler stated explicitly in his presentation, "…interoperability between systems standardization must be aimed at the modeling level, i.e., the standardized level must be higher than the programming level standards applied." This is a wise statement and indicates that for DTRA to succeed they must emphasize the coordination of the underlying conceptual models and harmonize the resulting operational ideas. Special interest should be taken to focus solely on standardizing the information exchange requirements but also to look at the modeled cause-effect-chains, which must be coordinated.

The intent for the IWMDT is to achieve the Universal System level of integration, with enterprise wide shared hazard information. The current design provides the user a common GUI that interfaces with HPAC, IMEA and INCA (nuclear tool) with a common mapping background. If this is achieved than the recent success shown in Figure 20 of response times can be expected to drop further while the resolution of the results will increase. And, more importantly the integrated collaborative nature of IWMDT will provide much more accessible data and process information.

Requirement	1990 Desert Storm	2003 Iraqi Freedom	2004-? IWMDT Released
Pre-Strike Analysis of Targets	>6 hours	<3 minutes	<2 minutes
Post-Strike Assessment Prediction	>12 hours	< 1 hour	<30 minutes
Oil Fire Impact	>30 days	<6 hours	<1 hour

Figure 20. Hazard Prediction-IWMDT Approximations.

[25] In Conference at 2003 Fall SISO meeting, Orlando Florida

1. IWMDT-TS Diagrams

The following diagrams demonstrate the engineering effort to integrate Targeting Support (TS) functions with the Consequence Assessment (CA) functions internal to the IWMDT tool. The diagrams are from the perspective of a user using IWMDT, and accessing the TS modules with a requirement to pass data to the CA modules for transport and dispersion. The TS software installer verifies the presence of CA on initial load and establishes the relationships through path assignments and specified interface calls. Figure 21 is a standard operation involving a user who chooses to evaluate a building, bunker or tunnel through the use of IWMDT-TS modules. This software allows the user to interactively design the target and iteratively develop weapon-target solutions. Figure 22 and 23 are sequence diagrams for the transport and dispersion of agents generated through the attack solutions executed within IWMDT.

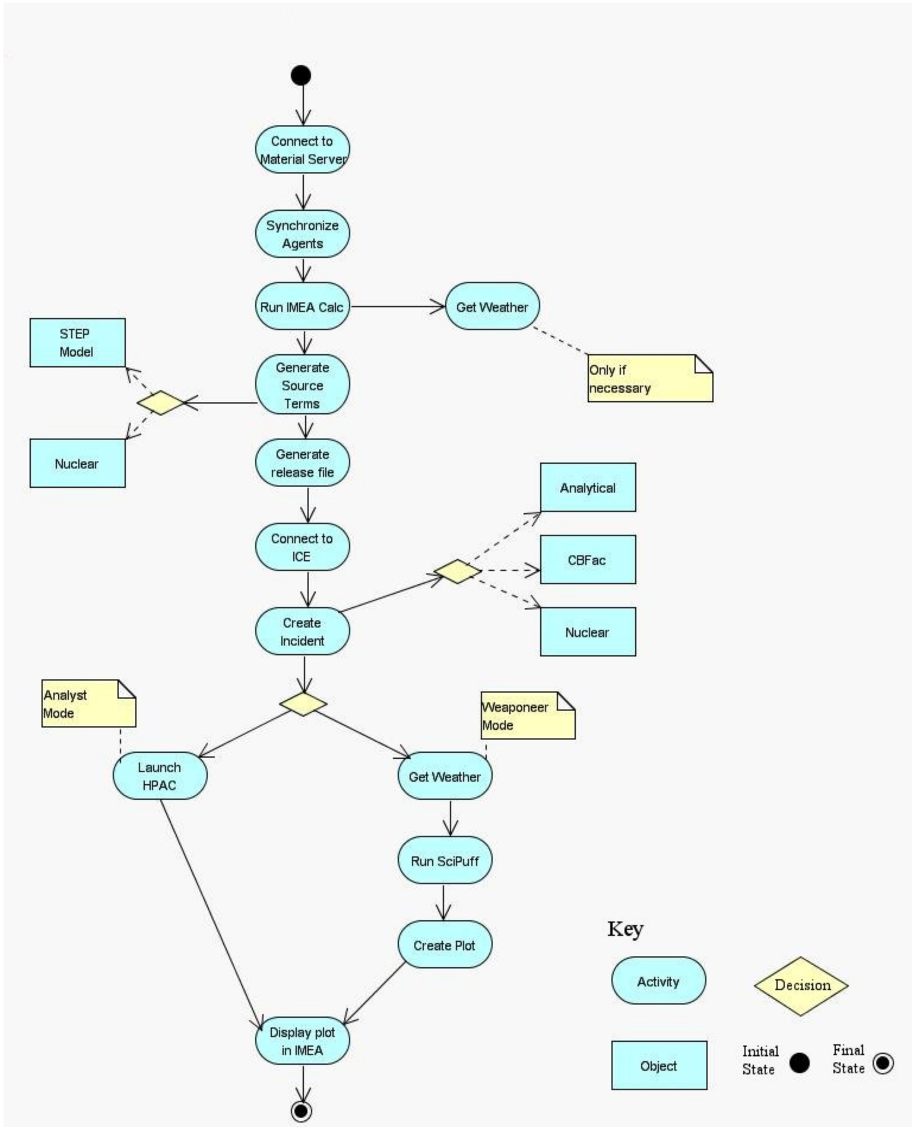


Figure 21. IMEA-HPAC Flow (From IWMDT Pgm Plan, 2003.)

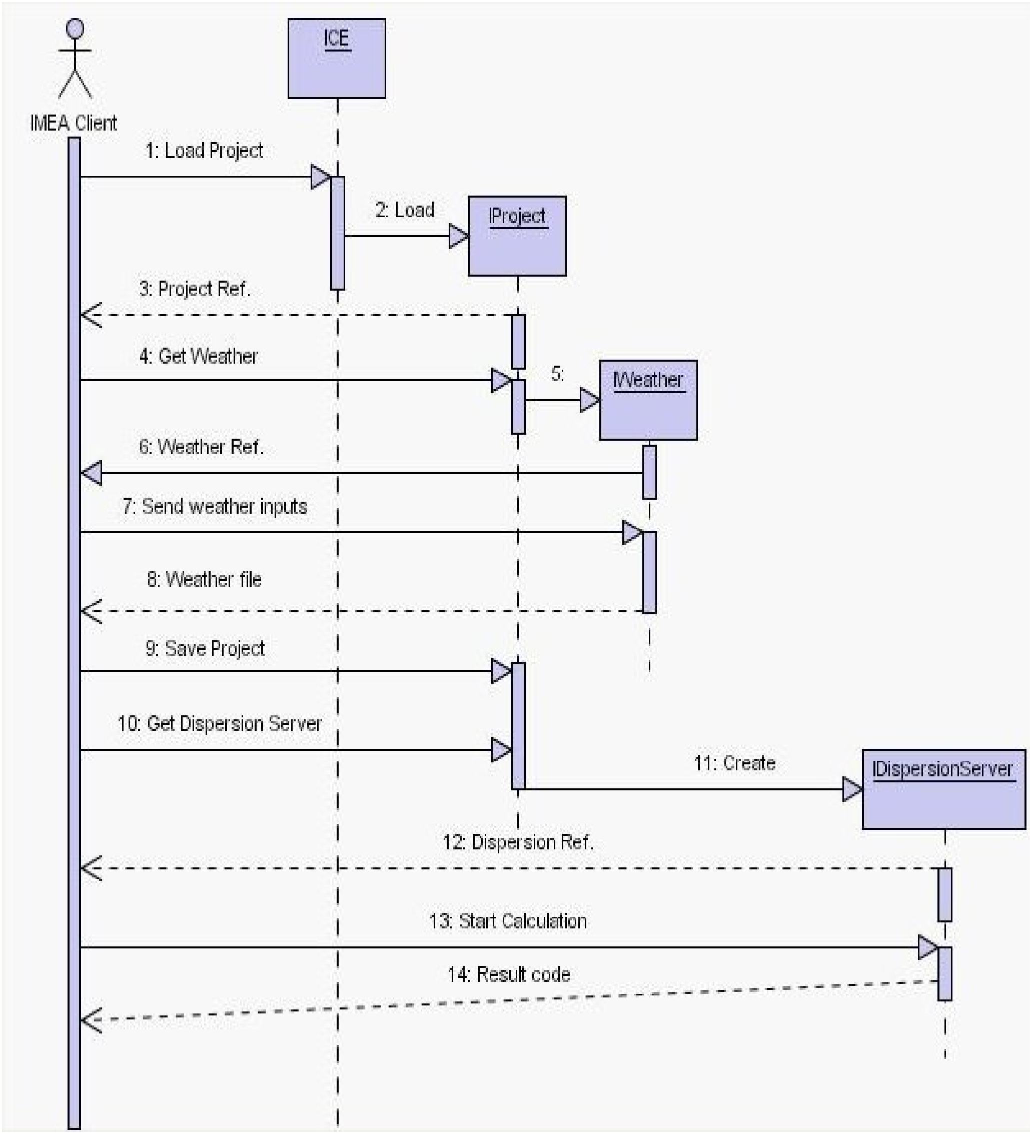


Figure 22. Sequence Diagram for SCIPUFF
(From IWMDT Pgm Plan, 2003.)

Sequence Diagram Activities as demonstrated above:

1. The **IWMDT** client connects to ICE to load the project file.
2. ICE creates a reference to the IProject interface.
3. IProject loads the requested project and returns a reference to the project interface.
4. The client requests a reference to the IWeather interface.
5. IProject creates an instance of IWeather.
6. IWeather is a response of the form.
7. The client populates the weather fields and sends them to the IWeather interface.
8. IWeather returns a weather file.
9. The client saves the weather to the project.
10. The client requests a reference to IDispersionServer.
11. IProject creates the server.
12. IDispersionServer returns a reference.
13. The client initiates an asynchronous Scipuff calculation. During the calculation, the client polls Scipuff for the calculation status and updates a progress bar. If an error occurs during the calculation, the client displays the error message and returns control to **IWMDT**.
14. If the calculation completes successfully, IDispersionServer returns a result code.

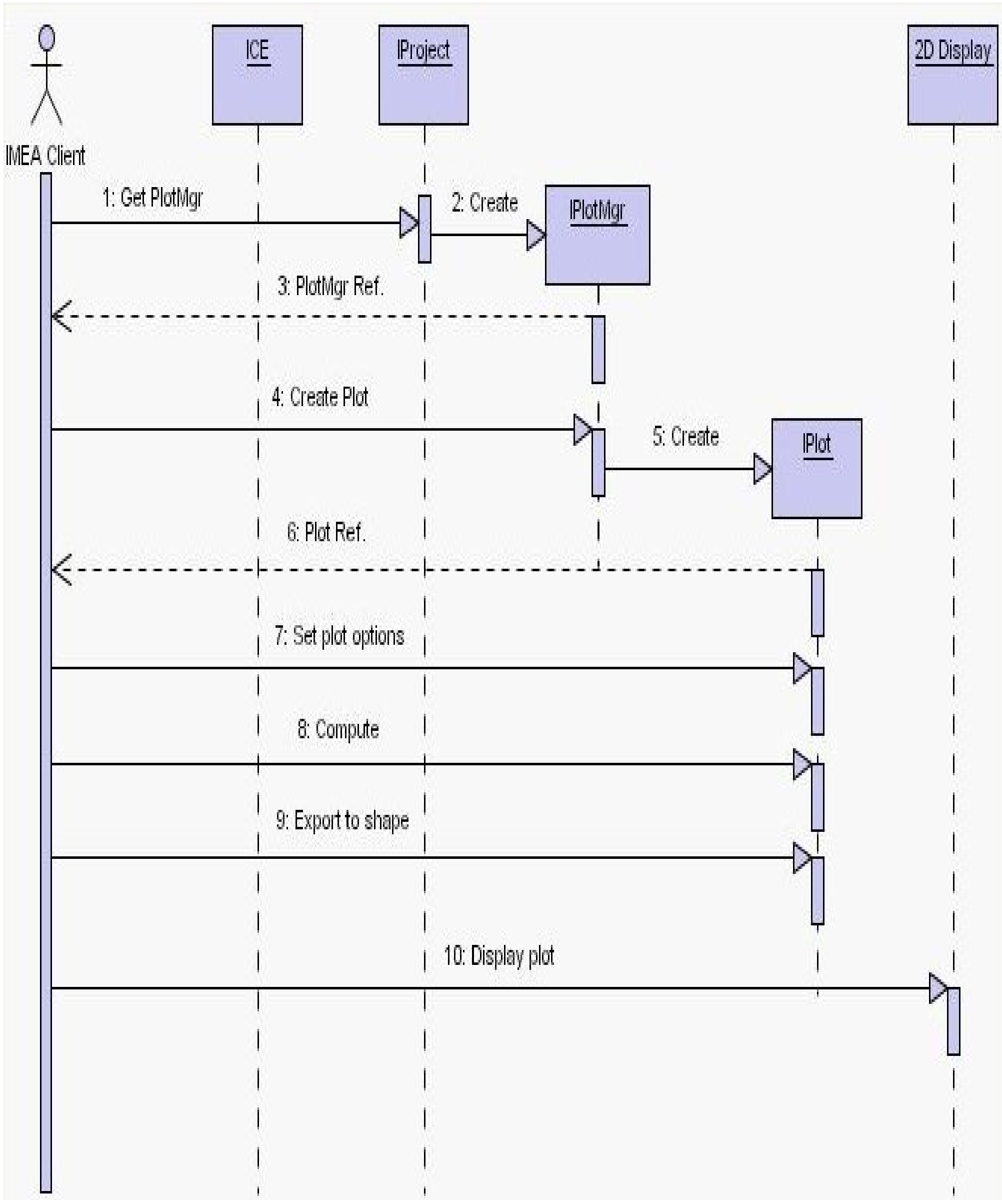


Figure 23. Generating a Plot .
(From IWMDT Pgm Plan, 2003.)

Sequence Diagram Activities as demonstrated above:

1. The **IWMDT** client requests the IPlotMgr from the IProject interface.
2. IProject creates an instance of IPlotMgr.
3. IPlotMgr returns a reference.
4. The client requests a reference to IPlot through IPlotMgr.
5. IPlotMgr creates an instance of IPlot.
6. IPlot returns a reference.
7. The client sets the plot options according to the options set by the user in the Plot Options dialog.
8. The client calls IPlot::compute to generate the plot.
9. The client calls IPlot::export to export the plot to a shape file.
10. The client opens the exported shape file.

V. IWMDT PRODUCT MANAGEMENT

A. IWMDT CONFIGURATION MANAGEMENT

The bane of most programmers is the requirement to maintain consistent and thorough documentation. In most cases it is not true to say that they consider the documents to be a waste of time, developers thrive on well-documented requirements. It is only to say that it is a nuisance to the creative flow to maintain blasé descriptions of their excellence. Couple this predilection for focusing on the code with the aggressive scheduling that most government leads establish for the programs, we often find that documentation is neglected.

As is true with most prototyping efforts, IWMDT aggressively developed a prototype first and is now establishing the structure of management. For IWMDT the existing management frameworks in place for the stand-alone integrated tools dictate much of the structure. The politics of funding and priorities for this project are incestuously linked with the previous programs. There are no less than four separate program managers that affect the development of IWMDT. Each of these program managers is responsible for continuing the stand-alone development while ensuring that they do not impinge on the success of IWMDT. To accomplish this task a number of efforts are emplaced to assist the program managers as well as the developers to maintain a consistent and integrated view. The first is the use of formal configuration management.

1. Configuration Management Definition

Configuration Management is often spoke about in the Government but rarely implemented with vitality. The primary reason why most government Contract Officer Representatives (COTR)[26] diminish the utility of CM is their misunderstanding of the process.

[26] http://arc.publicdebt.treas.gov/DWP/fs/fsrgprct.htm ,Jun 2004

The CM process is intended to address seven primary areas according to Carnegie Mellon, Software Institute, a leader in software configuration management. The areas are:

- **Identification**: identifying components, structure
- **Control**: controlling releases and changes
- **Status accounting**: recording, reporting status
- **Audit and review**: validating completeness
- **Manufacture**: managing construction, building
- **Process modeling**: ensuring life-cycle model
- **Team work**: controlling team interactions

Carnegie Mellon University as well as all other professionals in the field imply that these principles work when the process is routinely applied and institutionalized into the corporate process. Consistent with this intent is DTRA's adherence to IEEE/12207. According to the IEEE 12207 definition shown below the adoption of this standard is different in content but consistent in intent with commercial definitions as expressed by Carnegie Mellon.

The IEEE/12207.2 definition is: identify and define software items in systems, record and report the status of the items and modification requests, ensure the releases of the items, ensure the completeness consistency, and correctness of the items, control storage, handling, and delivery of the items.

There are other consistent perspectives that illustrate the consistency of intent while applying differing focuses. The two statements below further illustrate this point.

Software Configuration Management involves identifying the configuration of the software (i.e., selected software works products and their descriptions) at given points in time, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the software lifecycle. The work products placed under software configuration management include the software products that are delivered to the customer (e.g., the software requirements document and the code) and the items that are identified with or required to create these software products (e.g., the compiler). **The SEI Software Capability Maturity Model (version 1.1)**

The key role of Software Configuration Management (SCM) is to control change activity. If, however, SCM is viewed merely as a management tool or contractual obligation, it can easily become a bureaucratic roadblock that impedes the work. While such systems may be contractually required, the real need is to assist the programmers in controlling and tracking their work, while ensuring that nothing is lost or destroyed. **Watts Humphrey; Managing the Sw Process; Addison-Wesley, 1989**

Defining the process is the beginning of understanding, implementing it is the proof of understanding. To accomplish this DTRA established a top-level working group tasked with the responsibility of ensuring the CM for IWMDT is consistent and appropriately applied.

2. PIWG Directed CM Processes

DTRA instituted a Product Integration Working Group (PIWG) to oversee the configuration management of all Technology Directorate software applications; this group is responsible for the top-level CM for IWMDT. The body is organized in a series of working groups tasked with researching and reporting on specific areas of interest to the PIWG. According to the DTRA IWMDT Prototype CM Plan Rev 2 the IWMDT system is under the CM control of the PIWG. The following directives and responsibilities associated with IWMDT are extracted from the CM Plan.

The DTRA/TD M&S Master Plan and the IWMDT Program Plan establish the PIWG as the primary body for; reviewing technical requirements, recommending policy, developing processes, and overseeing implementation of the IWMDT. One of its duties is to serve as the Configuration Control Board (CCB) for IWMDT, managing the configuration of the IWMDT and reviewing all requested changes to configurable items. The PIWG determines and approves all schedules, policies, procedures, and directives relating to CM.

PIWG duties also include the following; determination of baseline version release, determination of installation sites, approval of requirement additions, deletions, or changes Change Requests, Problem Reports, and design change requests (PRs, CRs, DCRs). Within the CM task are a series of supporting tasks including, approve CM-controlled documents and document changes, and approval of the Verification and Validation (V&V) Plan. Because approving fixes in the IWMDT product must occur as problems are discovered, the PIWG may approve development work via Email. Regardless of the method there are distinct areas that the PIWG is responsible for controlling.

3. Software Configuration Items

Under the PIWG three broad areas were identified to manage IWMDT products. The three configuration item types, which the CM process controls, are: software items, documentation items, and miscellaneous items.

In accordance with working level IWMDT documentation, the first type, software items is described as shown below. Software items consist of three categories of software: (1) COTS software, such as web servers, database engines, etc. that provide generic functionality to the IWMDT, (2) externally-developed software (developed outside of DTRA or within DTRA but separately of the IWMDT) such as MEA, HPAC, and INCA, that provide WME-specific functionality, and (3) software developed by/for DTRA specifically to provide WMET-specific functionality as a web-based application.

Because much of the software under the IWMDT architecture is either category 1 or category 2, which is externally maintained, the IWMDT CM plan only covers category 3. Appendix C contains a list of all Computer Software

Configuration Items (CSCIs)[27] that make up the IWMDT software. To facilitate the concurrent development of disparate pieces of IWMDT, PIWG has directed multiple baselines may exist simultaneously.

Figure 24 displays a notional relationship of the simultaneous baselines within the development activities and the CM process. The CM Plan specifies that the PIWG will make a determination as to when new features are to be frozen, and the developers will only work on fixing problems and finalizing a release. Following a formal beta test where a software beta version is numbered (e.g., Version 5.3 Beta 1), the PIWG finalizes the expected release date, and determines which PRs are to be fixed by the release. For example, the PIWG may request that all PRs associated with certain functionality be fixed unless specifically rejected by the PIWG prior to acceptance.

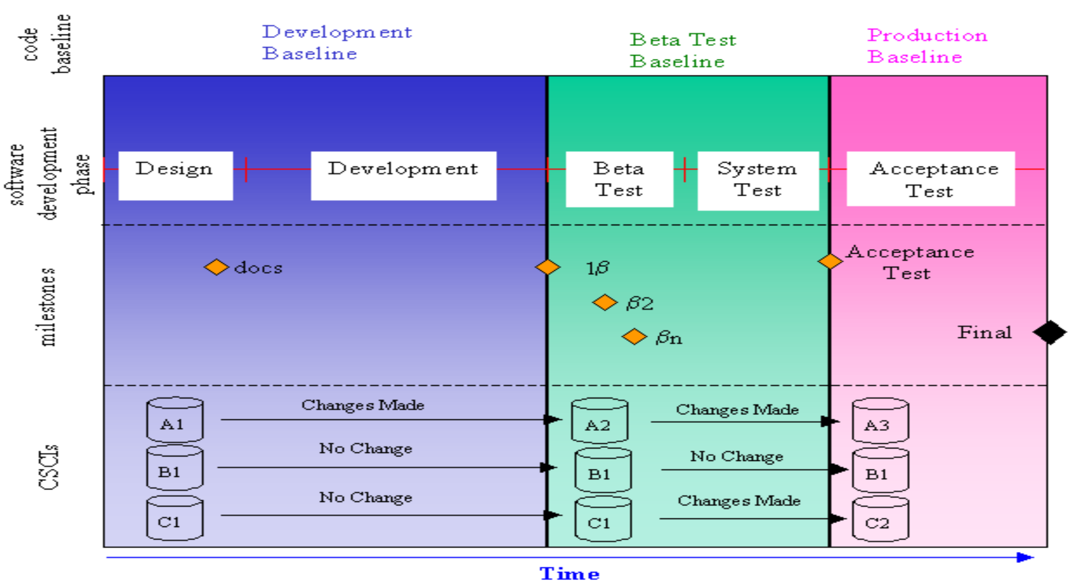


Figure 24. Flow of Configuration Item Data in Baselines. (From IWMDT Pgm Plan, 2003)

[27] Appendix C of this document

B. VALIDATION, VERIFICATION AND ACCREDITATION (VV&A)

The field of validation and testing is a very broad field with a myriad of standards and methods. Across this myriad is a common thread, which generally asserts that Models and Simulations (M&S) credibility is measured by the V&V process and approved through the accreditation process. Employing these processes serves a number of purposes. Select reasons to employ VVA are: it provides increased confidence in the model, reduces the risk of using the model, it provides developers a means to contain cost, potentially it provides better analysis and it can satisfy policy requirements by expanding users training activities. Though the IWMDT is only a prototype and is not in compliance with many of the prescribed DoD standards or methods, they are more concerned with sound engineering than most of the previous efforts.

Appendix A is the Department of Defense INSTRUCTION NUMBER 5000.61, dated May 13, 2003, which addresses DoD Verification, Validation and Accreditation. Within this instruction are some very important directions, which address the authority and responsibility of Component Commands, and DoD sponsors. Additionally it instructs that VVA be incorporated into the life-cycle management of all models and simulations commensurate with their relative importance and risk.

Much of this thesis highlighted the risk of linguistic discontinuity when discussing terms and concepts. Our research indicates that it is not linguistic discontinuity but ignorance of the definitions that cause improper use of the three key words discussed in this section. The three key words; “validation”, “verification” and “accreditation”, each have a very clear definition. And this clear definition is often confused with “correct”, which can be described in terms of the ability of a code block to hold to specified pre-conditions.

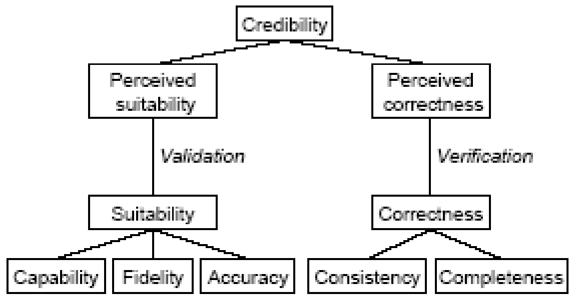


Figure 25. Dependencies between V&V related terms.[28]

According to Neil Storey [29], validation is the process of determining that a system is appropriate for its purpose. This is an important definition that is very often misstated and misapplied by users and software experts. This is not a case of linguistic discontinuity; this is a case of incorrect use of the phrase. The incorrect usage most often used is "testing the software to ensure it does not crash" sic. This is neither a proper use of the term or an accurate process that is repeatable. A more accurate application of the term would be ensuring that an application conforms to a specified level of accuracy when its outputs are compared to an aspect of reality. The key to validation is the appropriateness of the software not the functionality of the system. The solution is not a measure of "goodness"; it is only a measure of the difference between the model and the real world.

Neil Storey's definition of verification is the process of determining that a system, or module, meets its specification. It should be noted that the key in verification is not the appropriateness but the functionality. As is true with the incorrect use of validation, verification is incorrectly used with the same definition, "testing the software to ensure it does not crash" sic. The key here is that the software meets all the specifications as stated.

[28] "Generalized Process For The Verification And Validation Of Models And Simulation Results" , Dirk Brade, 2004

[29] "Safety-Critical Computer Systems", Addison-Wesley, 1996, ISBN: 0-201-42787-7

An example is “create folders on the IWMDT server <SRS-CA-16-d><IWMDT CA-VT-197-d>”. If the system is able to create a folder on the server it has met its specification. If the software is not capable of meeting the specification it is fails the verification and must be corrected prior to release.

The final definition of the three is accreditation, Storey does not specifically refer to this term, and instead he addresses certification. The difference is subtle so we include certification definition first to allow us to be consistent with Storey on all three terms. Storey defines certification as convincing an external regulating body that the system is safe..”. According to DoDD 5000.59 accreditation is “the official certification that a model or simulation is acceptable application”. The process is not to convince the regulating body that it is safe, the focus is on convincing the body that it meets its original specifications. Ideally the certifying official should be involved in the process as early as possible to better understand the requirements. Because the accreditation is based on the specific application of the application, it is possible that the code could be accredited for one use and not valid for a second use.

C. SYSTEM AND DATA SECURITY

The security of the data should be viewed from a number of perspectives. On the most elementary level is data integrity. According to Storey[30] , data integrity can be defined as: *the ability of a system to prevent damage to its own database and to detect, and possible correct, errors that do occur.* Data integrity is always important but if the user is not familiar with the expected results the integrity of the data is much more critical. Operators of IWMDT (or the predecessor stand-alone components of IWMDT) rarely understand the agent definition or algorithm process; they merely understand how to use the tool. Because users usually understand only the tool use, they do not have an experienced perspective on acceptable results. The users are not able to consistently identify data errors or potential data

[30] “Safety-Critical Computer Systems”, Addison-Wesley, 2001, ISBN: 0-201-42764-7

inconsistency without this strong domain background. This lack of background is likely to cause the typical user to be unable to create unique material definition files. The typical user is a first-responder or warfighter whom may be familiar with the tool but knows very little about the dynamics of agents in the atmosphere. The media has so mystified the terms associated with WME that most people are not familiar with the basic elements associated with WME assessment.

According to source documents DTRA is only performing minimal security effort. Because it is a prototype with a sure future it is important to follow as many security standards as possible, but not enough to slow down the prototype process. Initial security effort is limited to[31]: partial DITSCAP- only prepare for certification and accreditation, execute security readiness review (SRR) scripts on IWMDT baseline, provide security guidance on ‘as solicited’ basis, and encourage active team participation in all aspects of information and procedural security awareness. This issue will be addressed in future development efforts as funding is provided and DoD guidance is clarified.

[31] DTRA IWMDT Internal CM Plan 2004

THIS PAGE INTENTIONALLY LEFT BLANK

VI. DISCUSSIONS AND CONCLUSIONS

A. PERCEIVED INCONSISTENCIES

The author’s experience has taught him that if we want reams of feedback the best time to gather it is not when we are trying to solve a problem, but instead shortly after we have made our decision. Everyone is a critic after one has made a decision, yet there always seems to be a void of good ideas when one needs them to make the decisions. In many ways this continues to be true in the development of the IWMDT architecture. For over a year as the pleas for input were openly requested the masses seemed quiet, but as the development approaches a release, the critics are many.

Not to complain, any productive comment is always useful, but not always appropriate for the place and time that one may receive it. The continued migration of source code and introduction of new system variables is introducing potential risk for faults in IWMDT. Users and development partners because of the dormant nature of much of the introduced code and algorithms still hotly contest much of the perceived “fault” introduction in this HPAC that is carried forward into IWMDT. As a direct factor of the complexity of the code it is highly likely that two trained users will get different answers in the same exact scenario because of minor procedural techniques. Though seen as a fault by many because it is perceived to have dormant faults that could cause errors that are not easily identified or traceable by the user, others see it as a flexible implementation of necessary variables.

Obviously because a code allows users to enter inconsistent data does not automatically classify it as unsafe. The dormant fault and potential error certainly is represented in the definition “fault” but because it operates correctly if correct information is entered it is not a system fault. Further complicating the reengineering effort was an understanding that this tool would be used by different countries, military, first responders, city through national government agencies and academia.

The broad user base introduces different safety-criteria both from an assessment requirements perspective as well as from the operational perspective. Therefore, the HPAC engineers had to consider the applicable differences in their analysis of the system design.

To address these issues and many others, DTRA continues to conduct scores of user reviews, user testing events and independent live-fire tests and wind tunnel test to validate various pieces of the code and operational model. As the code becomes more broadly accepted and users are better educated arguments illustrating “faults” are rapidly replaced by the naysayer with the charge that the code design is unsafe.

Some adversaries even claim that the design encourages unsafe operator errors as a result of the GUI design openness. In response to the “openness” of the GUI, DTRA developed a series of defaults that improved the system integrity by providing “smart defaults”. The “smart defaults” were set up on a template basis, which changes color on the screen if modified. This enables the operator to easily reset his factors if modified to an approved and tested weapon-target set of data.

The concept of “failsafe” entries is used to a lesser degree prohibiting impossible combinations of variables, while still allowing implausible combinations. The engineering logic used to develop the smart defaults and the limited “failsafe” entries is sound represents much research, live field tests, and controlled experiments. The debate will continue, the best path forward is to continue to validate the results against reliable data, provide quality assessments and work with the users for evolving requirements.

Before discussing specific aspects of the architecture and performance it is important to note that although IWMDT is designed in accordance with the users’ requirements, and the software is functional within the intended environment, there will still be implementation issues. In some Commands, it is possible that the local security constraints or local common operating picture (COP) software may refuse to share data that is critical for IWMDT. Understanding this possibility, IWMDT is intentionally establishing robust data storage areas to provide SMEs forward a repository of approved models and templates. This is an important point for the developer to consider when assessing the architecture. The ability of the user to implement our design even if correctly designed may still be an issue in many instances.

B. GENERAL GUI DISCUSSION

Because DTRA products enjoy a degree of familiarity among most CBRNE users, it would be assumed that IWMDT navigation would take advantage of the familiarity. But the author did not find that to be the case. The author who is familiar with the previous tools, web page delivery, CBRNE information and data access was still confused when first using the tool. Many suggestions are currently being reviewed to improve the interface. One of the leading options is the standardization of all screens with one format. Though this will have an immediate benefit of assisting the user with consistency it will undoubtedly force the navigation for some tools. Not all the tools that are currently integrated and certainly not the future ones not scoped will be capable of using one interface. This is an issue that must be story boarded and a technical solution must be found before this project goes further. It is imperative that the user not be forced to reduce his capability only to meet an interface standard. The full strength of integrated capabilities must be exercised. The creative GUI must accommodate this growth.

The current navigation flow encourages users to move right to left and top to bottom, but this is not necessarily intuitive. The initial expectation is that the user is familiar with previous tools and they will intuitively move from right to left and top to bottom, with some minor coaching this probably is not a large leap of faith. The designers took special effort to ensure that they provided a capability-based approach, which encourages users to use similar processes from other aspects of their jobs that involve similar processes.

By taking a capability-based approach users may create their own customized web layout that facilitates quick access to IWMDT capabilities. Providing this option may alleviate many of the concerns over the GUI format. The initial operational capability (IOC) is focused on weaponeering and consequence assessment capabilities, with additional focus on traditional tool-based usage facilitated over a web interface.

The IWMDT concept of support is shown in Figure 26, the key to the diagram is the reliance on the definition of rules for the components of the IWMDT to exchange data and interface IWMDT with external (the "Rest of the World") capabilities.

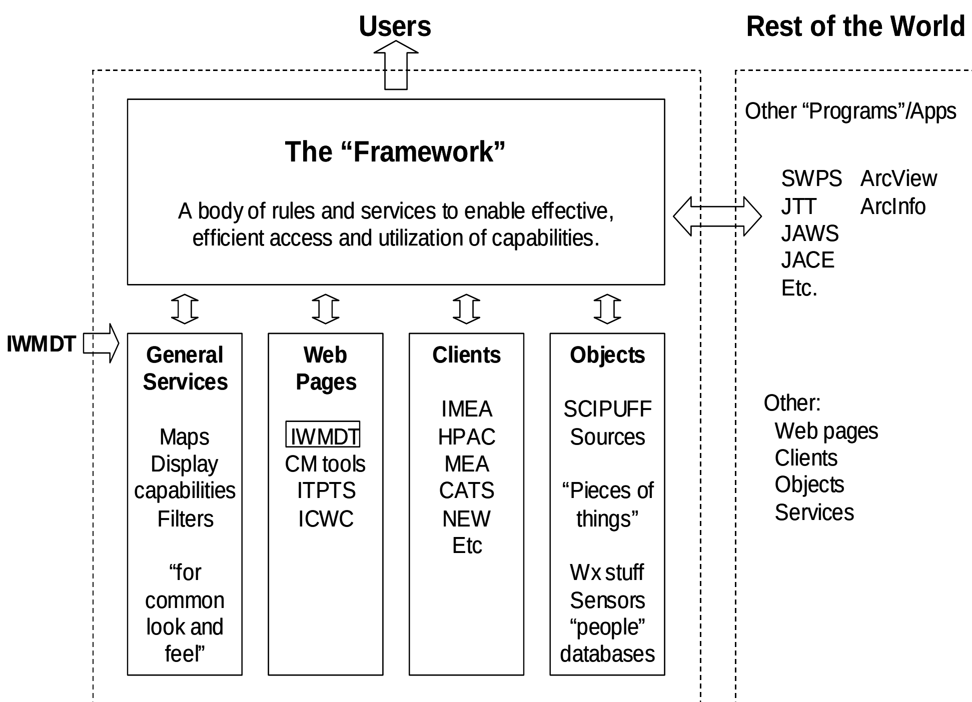


Figure 26. IWMDT Concept of Support.
(FROM: IWMDT How-To Manual, 2004)

C. COE DISCUSSION

IWMDT uses the following COE Version 4.7 Build Lists[32] for Windows 2000 and Solaris 8 specifications. These formats are specifically used to comply with the COE system requirements.

- Java 2 Standard Edition (J2SE) Java Development Kit (JDK) version 1.4.1
- J2SE Java Runtime Environment (JRE) version 1.4.1
- Perl
- Tcl/Tk
- Tivoli
- Informix
- Microsoft SQL Server (Windows 2000 only)
- Oracle RDBMS
- Sybase
- Apache HTTP Server
- Jakarta Tomcat Servlet Engine
- BEA WebLogic
- Netscape Directory Server
- Netscape Enterprise Server
- Netscape Browser
- Websphere
- Joint Mapping Toolkit (JMTK)

The COE guidelines are strict in respect to build list use. In all cases where the functionality is required and software on the build list meets the requirement, the software must be used. Despite this strict inclusion policy and the call for use of the JMTK, DTRA chose not to use the JMTK. Instead they are using the commercial JMTK as noted earlier in the thesis in order to be commercially viable and more visible.

D. DEPLOYMENT OPTIONS

Though the primary deployment method envisioned for IWMDT is through the use of a web-browser as the GUI and remote databases as the source data, other methods are expected. The method with the most promise for continued use is through the shared web-services approach. This approach is not a manual user

[32] Available online at http://www.disa.mil/coe/coeeng/RELEASE_PAGES/Ver47pg.html.

method using a GUI; it relies on industry standard interface exchanges through the various web-services technologies. The author is confident that the greatest benefit to the warfighter and first responder community will be the use of this method. In most cases the stand-alone HPAC, IMEA and nuclear tools applications may remain the best way for many users to execute the CBRNE mission. For many reasons including the local control, not requiring on Internet connectivity, local speed and many other factors, the current standalone will remain a viable method.

Implementing a distributed method of installation indicates that some portion of the storage or computational effort is performed by a different system. Standalone operation means that all necessary computation and capabilities are available on the local machine that is required for operation of the software. In both cases the external use of weather is preferred in some instances, this is not indicative of method type as it is not a requirement.

1. Distributed

IWMDT is designed for accessibility through a variety of distributed combinations. These combinations are distinguished by the burden of requirement on the client. The browser client configuration shown in Figure 27 has the lowest requirement on the client. The only requirement on the client is to have a standard web browser running on the local machine.

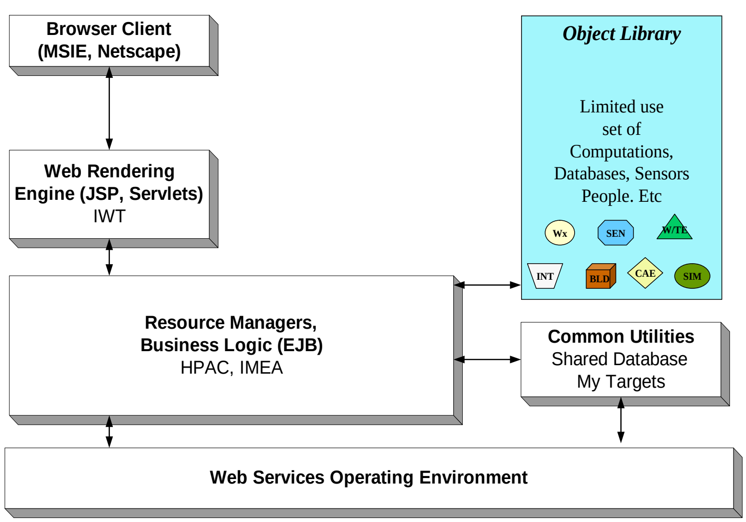


Figure 27. Web browser distributed deployment.
(FROM, IWMDT Project Management Briefings, 2003)

2. Local/Distributed Combination

A second option is allowing the user to install application clients on their local machine. In this configuration the clients access computational models either installed on the same machine as shown in Figure 28 or installed on a remote server as shown in Figure 29. The use of the same business logic, resource managers and computational object library as the browser-based client configuration is a core requirement of the distributed methodology. Identical processes ensure the necessary consistency of the tool and the integration of databases. Additionally it allows task sharing between the local and remote processors if required by the user. The standalone application cannot inadvertently allocate and monopolize an item from the object library.

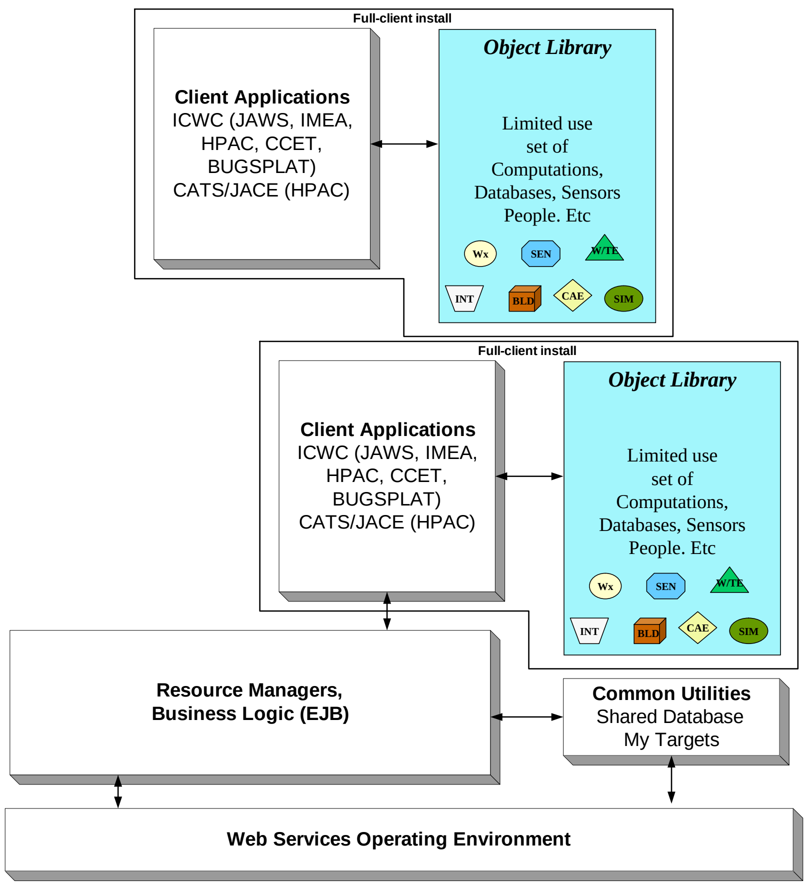


Figure 28. Application client accessing local objects, remote database. (FROM, IWMDT Project Management Briefings, 2003)

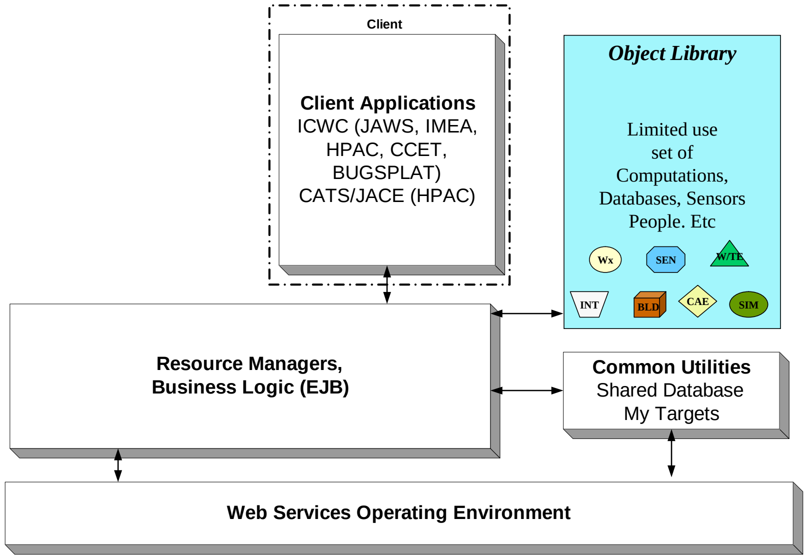


Figure 29. Application client accessing remote objects. (FROM, IWMDT Project Management Briefings, 2003)

3. Standalone

As stated earlier, this configuration relies on the total required code be installed on the local machine without a requirement for external processes. This configuration causes extreme requirements on the local machine for space and processing ability. The local storage requirement is quite often as high as 4GB of programs and data, and escalates for additional terrain and map files. Though it has the greatest burden on the use, it provides the greatest flexibility. Without relying on external processes the system is functional virtually anywhere in the world. Once again the processes are performed locally, the databases may be accessed remotely if required.

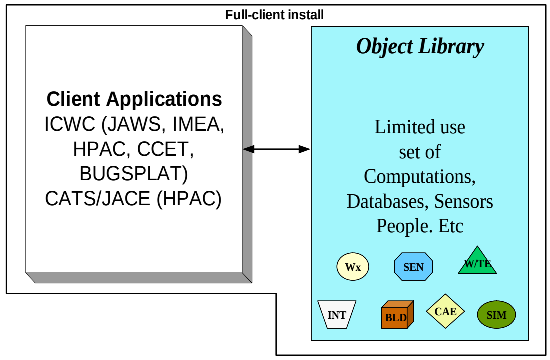


Figure 30. Standalone installation.
(FROM, IWMDT Project Management Briefings, 2003)

E. IWMDT ALIGNMENT WITH HF INITIATIVE

The greatest measure of success for the IWMDT is the ability to integrate their services and data into a shared global data infrastructure. DTRA has chosen to accomplish this task through the integration of IWMDT into the Horizontal Fusion effort. The author created the following chart to highlight the current status of the IWMDT program as it pertains to the HF initiatives. All the data in the chart is elicited in one of three documents distributed by the HF team.[33] The evaluations are largely the opinion of the author with limited input from the IWMDT development team. This is not representative of the IWMDT program manager's assessment.

John Steibet, the CIO for Assistant Secretary for Defense, Network and Information Integration Office, (ASD/NII) advanced the HF portfolio. The goal is to integrate and optimize technology and operations to achieve "Power to the Edge" in the new battlespace. On the next page is how the chief architect John Osterholz describes this new opportunity[34]:

"The difference between previous attempts at interoperability and now is that – with emergent technologies – global reach and rapid movement of critical battlespace knowledge mean that true interoperability is actually possible."

---

[33] http://horizontalfusion.dtic.mil/, Jun 2004

[34] Horizontal Fusion Initiative document, Feb 2004

According to source documents, Osterholz claims this is possible because of many of the technologies discussed in this thesis, but additionally hardware and tool partners the portfolio is integrating. The Horizontal Fusion Portfolio Initiative was undertaken by DoD's Office of ASD/C3I/CIO to accelerate the transition of Net-Centric War fighting from vision to reality.

The key governing imperatives to success are identified by the Portfolio Manager as; Horizontal Fusion is focused on the building of a services-oriented architecture which supports DoD Operations (SIPRNet), core enterprise services must be leveraged, MANDATORY use of Net-Centric Enterprise Services (NCES) Security Services (NSS) to authenticate users as per the specification, MANDATORY to label all data/XML/SOAP requests with relevant classification information, and MANDATORY to restrict access (where applicable) based on that user's attributes.

DTRA is currently aligning their software development practices to adhere to the principles of Horizontal Fusion, with a goal of integrating with the Portfolio in 2005. Accomplishing this task requires DTRA to address the broad requirements established by the ASD/NII office for inclusion in the HF future development and integration efforts. Specifically HF requires planning and integration of existing projects to ensure a smooth integration with the Global Information Grid (GIG). By integrating tools and providing an open architecture, DTRA enables the posting and processing of data in a more timely and consistent manner. The key to this integration is the definition and integration of the web-services in compliance with the intended systems. The author staffed the following Statement of Work amongst the IWMDT team and presented it to the Government for official action. Within the narrative is the real analysis of this tool and the applicability to other programs. If accepted for entry then the items not addressed above must be addressed to make this a viable web-service oriented applications. Figure 31 shows the concept overview slide (OV1) for the existing IWMDT application framework. This overview demonstrates the utility of this thesis and the development of the IWMDT.

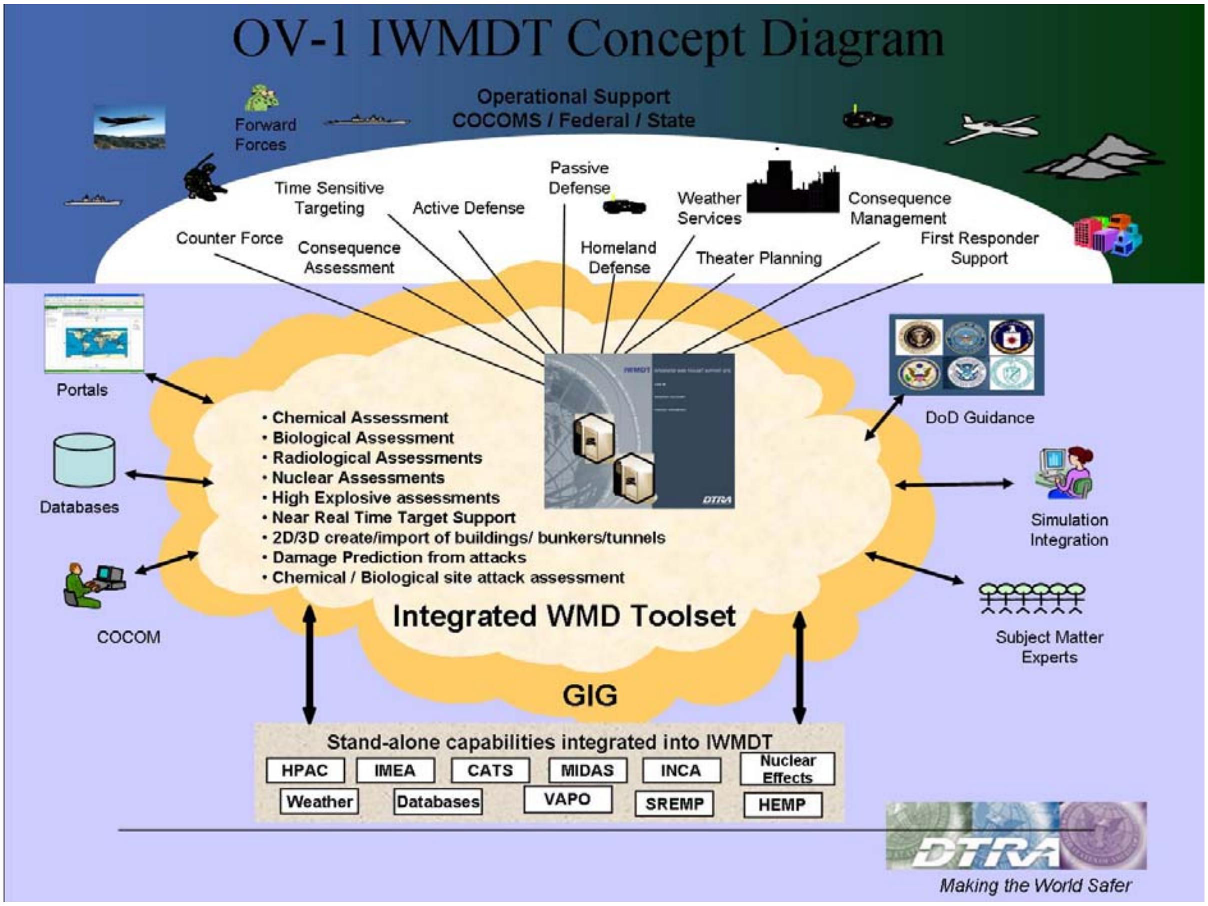


Figure 31. IWMDT OV-1 Diagram

F. CONCLUSION

Vision is the key to innovation, and DTRA has demonstrated enough vision to develop a prototype that will be capable of meeting existing and projected CBRNE requirements. The veracity of their claims and the resoluteness of their focus will be borne out in the next six to twelve months. Over this period the program will either prove its technical merit or it will fail and DTRA will attempt another solution. Over the period of writing this thesis, a myriad of engineers, and managers have influenced the IWMDT effort, too many to capture. But the future will not be defined by the individuals it will be defined by the teamwork of a small handful of government leaders and contractors with the vision and skills to develop on the edge of technologies and emerging requirements.

The IWMDT effort represents the cooperative engineering approach of over 100 separate engineers and managers with limited budget, loosely scoped requirements and uncertain support, to this end the effort is already a success. This has been accomplished and the resulting applications and integration capabilities will emerge in the near future. The technical goal was to meet deployable CBRNE users requirements in their daily preparation, execution and post analysis of CBRNE threats, within the next twelve months this will be tested and the author is confident it will be measured effective.

Through the innovative web-services based approach and the disciplined adherence to standards and common formats, this program is postured to succeed in many areas. These areas will commonly require rapid, accurate and accessible data and specifically tailor the tool to meet their other unique requirements. The future of CBRNE assessment looks a lot like IWMDT, though the cover may change, the book will read the same.

APPENDIX A. DOD INSTRUCTION 5000.61

Department of Defense
INSTRUCTION

NUMBER 5000.61
May 13, 2003

USD(AT&L)

SUBJECT: DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)

References: (a) DoD Instruction 5000.61, "DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)," April 29, 1996 (hereby canceled)
(b) DoD Directive 5000.59, "DoD Modeling and Simulation (M&S) Management," January 4, 1994
(c) DoD 5025.1-M, "Department of Defense Directives System Procedures," March 5, 2003
(d) DoD Directive 5141.2, "Director of Operational Test and Evaluation (DOT&E)," May 25, 2000
(e) through (p), see enclosure 1

1. REISSUANCE AND PURPOSE

This Instruction:

1.1. Reissues reference (a) to implement policy, assign responsibilities, and prescribe procedures under reference (b) for the verification, validation, and accreditation (VV&A) of DoD models and simulations and their associated data.

1.2. Authorizes publication of DoD 5000.61-G, "DoD Verification, Validation, and Accreditation Guide," consistent with DoD 5025.1-M (reference (c)).

2.1. The Office of the Secretary of Defense (OSD), the Military Departments, the Chairman of the Joint Chiefs of Staff, the Combatant Commands, the Office of the Inspector General of the Department of Defense, the Defense Agencies, the DoD Field Activities, and all other organizational entities in the Department of Defense (hereafter referred to collectively as "the DoD Components").

2.2. All models and simulations developed, used, or managed by the DoD Components after the effective date of this Instruction.

2.3. Models and simulations used in support of Operational Test and Evaluation (OT&E), all of which are subject to guidance from the Director, OT&E, per DoD Directive 5141.2 (reference (d)).

3. DEFINITIONS

Terms used in this Instruction are defined in enclosure 2.

4. POLICY

It is DoD policy that:

4.1. Models and simulations used to support major DoD decision-making organizations and processes (such as the Defense Planning and Resources Board; the Joint Requirements Oversight Council; and the DoD Planning, Programming, and Budgeting System (references (e) through (g)) shall be accredited for that specific purpose by the DoD Component M&S Application Sponsor.

4.2. Each DoD Component shall be the final authority for validating representations of its own forces and capabilities in common-, general-, or Joint-use M&S applications and shall be responsive to the other DoD Components to ensure its forces and capabilities are appropriately represented.

4.3. Models and simulations used to support joint training and joint exercises shall be accredited for that specific purpose by the DoD Component M&S Application Sponsor.

4.4. Accreditation requirements of models and simulations used to support all other applications shall be determined at the DoD Component level.

4.5. The DoD Components shall establish VV&A policies and procedures for models and simulations they develop, use, or manage.

4.6. Each DoD Component shall comply with the responsibilities identified in section 5. and procedures identified in section 6.

5. RESPONSIBILITIES

5.1. The Under Secretary of Defense for Acquisition, Technology, and Logistics shall:

5.1.1. In coordination with the DoD Components, develop policies, plans, procedures, and DoD issuances for the effective implementation and management of VV&A of DoD M&S.

5.1.2. Through the Director, Defense Research and Engineering, as Chair of the DoD Executive Council for Modeling and Simulation (EXCIMS):

5.1.2.1. Encourage improved communication and coordination among and between organizations and agencies conducting DoD VV&A activities.

5.1.2.2. Identify and support investments in VV&A enabling technologies that have high-value return in fulfilling DoD requirements, or that fill gaps in DoD VV&A capabilities.

5.1.2.3. Promote joint and cooperative research, development, acquisition, and application of VV&A technologies and processes among the DoD Components.

5.1.2.4. Establish standards and guidelines to promote DoD VV&A procedural commonality and foster M&S interoperability.

5.1.2.5. Arbitrate differences in representation of forces and capabilities among the DoD Components to ensure standardization in common, general, or Joint-use M&S applications and federations of models and simulations.

5.1.3. Designate the Defense Modeling and Simulation Office as the "DoD VV&A focal point" and the central source of DoD VV&A information.

5.1.4. Comply with responsibilities specified in paragraph 5.3.

5.2. The Assistant Secretary of Defense for Command, Control, Communications, and Intelligence shall:

5.2.1. Through the Director, Defense Intelligence Agency:

5.2.1.1. As the DoD Modeling and Simulation Executive Agent (MSEA) for M&S representations of foreign forces, for other DoD Components' representations of foreign forces, and their systems shall:

5.2.1.1.1. Serve as the final validation authority (reference (b));

5.2.1.1.2. Resolve validation issues; and

5.2.1.1.3. Be responsive to that DoD Component to ensure that foreign forces and capabilities are appropriately represented (reference (b)).

5.2.1.2. As the DoD MSEA for M&S representations of U.S. National and Joint Intelligence processes, for other DoD Components' representations of U.S. National and Joint Intelligence processes shall:

5.2.1.2.1. Serve as the final validation authority (reference (b));

5.2.1.2.2. Resolve validation issues; and

5.2.1.2.3. Be responsive to that DoD Component to ensure that intelligence processes and capabilities are appropriately represented (reference (b)).

5.2.2. Comply with responsibilities specified in paragraph 5.3.

5.3. The Principal Staff Assistants (PSAs) and the Heads of the DoD Components shall:

5.3.1. Plan and provide resources, as needed, to carry out functional VV&A responsibilities according to DoD Component priorities.

5.3.2. Approve DoD VV&A policies and procedures, and DoD Publications.

5.3.3. Ensure non-DoD M&S applications they sponsor comply with established DoD VV&A policies and procedures.

5.3.4. Establish VV&A policies, procedures, and guidelines for M&S applications and their associated data. DoD Component VV&A policies and procedures shall address, as a minimum:

5.3.4.1. Use of existing or new models and simulations, including those that are federates or federations.

5.3.4.2. DoD Component-managed models and simulations used for joint-, general-, or common-use applications.

5.3.4.3. Models and simulations used by the DoD Components that are developed, used, or managed by non-DoD organizations, (i.e., contractors (including federally funded Research and Development Centers), industry, academia, and other Federal or non-Federal government organizations).

5.3.4.4. Designation, authorities, and responsibilities of:

5.3.4.4.1. M&S Proponent(s).

5.3.4.4.2. M&S Application Sponsor(s).

5.3.4.4.3. Verification, Validation, and Accreditation Agent(s).

5.3.4.4.4. DoD Component M&S VV&A focal point(s).

5.3.4.5. VV&A documentation and accessibility requirements, as outlined in enclosure 3.

5.3.4.6. Application-specific data verification and validation activities that are included as an integral part of M&S V&V, accreditation, and documentation activities.

5.3.5. Establish procedures holding the following accountable and responsible for the activities indicated:

5.3.5.1. M&S Proponents:

5.3.5.1.1. Verification and validation of their assigned M&S, as well as the documentation of those activities.

5.3.5.1.2. Coordinating validation activities with the DoD Component who serves as the final authority for the validations of representations within its purview.

5.3.5.1.3. Funding the V&V over the life cycle (e.g., development, upgrades, and maintenance) of their models and simulations.

5.3.5.1.4. For distributed modeling and simulation or federations of models or simulations (hereafter collectively referred to as "federations"):

5.3.5.1.4.1. The M&S Proponent roles and responsibilities pertaining to V&V for the overall federation shall be fulfilled by the DoD Component organization responsible for managing a federation and its associated data.

5.3.5.1.4.2. The responsibility for V&V of a federate and its associated data shall be retained by the M&S Proponent for each federate within a federation.

5.3.5.2. M&S Application Sponsors:

5.3.5.2.1. As the Accreditation Authority, accrediting M&S used for their specific application(s), as well as the documentation of those activities.

5.3.5.2.2. Funding the VV&A activities that support their application-specific accreditation decisions.

5.3.5.2.3. Consulting with the appropriate MSEA during VV&A plan development if the models and simulations will involve representations within the domain of the MSEA.

5.3.5.2.4. Accrediting the federation and its associated data for the specific purpose shall be the responsibility of the DoD Component serving as the M&S Application Sponsor of a federation.

5.3.5.3. Individual Data Producers:

5.3.5.3.1. The quality of their data or data products provided for M&S use.

5.3.5.3.2. Supplying data quality information, including data verification and validation reports for data or data products provided for M&S use.

5.3.6. Designate a "Component VV&A focal point" to interface with the DoD VV&A focal point for their VV&A policies, activities, and documentation.

5.3.7. Document and make accessible to the other DoD Components the results of their VV&A activities, including, but not limited to, information and data on their DoD Component VV&A policies and procedures, V&V results, and accreditation decisions.

5.3.8. When designated as a DoD MSEA:

5.3.8.1. Upon request, provide domain information and expertise in support of VV&A activities.

5.3.8.2. Make certain that data quality information is available and accessible to support the individual DoD Component's VV&A activities.

5.4. The Chairman of the Joint Chiefs of Staff shall:

5.4.1. Establish VV&A policies, procedures, and guidelines to satisfy the needs of joint activities reporting to the Chairman of the Joint Chiefs of Staff.

5.4.2. In coordination with the other DoD Components, establish procedures for the validation and accreditation of joint M&S and federations of models and simulations used for joint applications.

6. PROCEDURES

6.1. Verification and validation (V&V) shall be:

6.1.1. Incorporated into the development and life-cycle management processes of all M&S.

6.1.2. Required for all models and simulations in current use in the Department of Defense.

6.1.3. Commensurate with the relative importance, risk, and life-cycle management phase of the model, simulation, or federation to which they are applied.

6.2. The V&V of a federation shall include a determination that:

6.2.1. Federation elements can physically connect and exchange data.

6.2.2. Federates, when joined together, provide adequate, accurate, and consistent simulated representations that adhere to the principles of fair fight and address the mission objectives.

6.3. Data V&V is an integral part of the M&S VV&A process and shall:

6.3.1. Be addressed, to include programming of V&V resources, at the earliest stages of a new model or simulation development or the upgrade of an existing model or simulation.

6.3.2. Be documented as part of the VV&A documentation requirements, as specified in enclosure 3.

6.4. VV&A information shall be documented and, as a minimum, shall include the information specified in enclosure 3.

7. <u>EFFECTIVE DATE</u>

This Instruction is effective immediately.

E. C. Aldridge, Jr.
Under Secretary of Defense
(Acquisition, Technology and Logistics)

Enclosures - 3
E1. References, continued
E2. Definitions
E3. VV&A Documentation Format and Accessibility Requirements

APPENDIX B. SPECIFIC IWMDT DATA FORMATS

Inputs	***Units***	***Limits***	***Defaults***
Name	N/A	50 characters	
Map	N/A	List of available maps	
HPAC Project Name	N/A	Available HPAC projects	(use internal project)
Notes	N/A	50 lines	
Attachments	N/A	Files, images, web page links	

Table 4. Sessions Inputs

Inputs	***Units***	***Limits***	***Defaults***
Name	N/A	50 characters	
Notes	N/A	50 lines	
Attachments	N/A	Files, images, web page links	

Table 5. CA Folders Inputs

Inputs	***Units***	***Limits***	***Defaults***
Name	N/A	50 characters	
Type	N/A	CBWPN, CBFAC, NWPN	CBWPN
Location	N/A	(See section)	N/A
Date/Time	N/A	(See)	N/A
Notes	N/A	50 lines	
Attachments	N/A	Files, images, web page links	

Table 6. Incident Inputs

Inputs	Units	Limits	Defaults
Munition	N/A	100 kg Bomb, 250 kg Bomb, 500 kg Bomb, Rocket, Aerial Spray, 152 mm Tube, 122 mm Rocket, 120 mm Mortar, Missile Bulk, Missile Subs, Ground Spray, Land Mine	100 kg Bomb
Delivery System	N/A	1 AC * 2 Drops * 4 Bombs, 2 AC * 8 Bombs, 4 AC * 8 Bombs, 1 AC * 48 Rockets, 2 AC * 48 Rockets, Battery (tube), Battalion (tube), Battery (rocket), Battalion (rocket), Mortar 36 Rounds, Mortar 72 Rounds, Aerial Spray, Missile Bulk, Missile Subs, Ground Spray, Land Mine	1 AC * 2 Drops * 4 Bombs
Agent	N/A	GA, GB, GD, GF, HD, VX, TGA, TGD, TGF, THD, TVX, ANTH, BOTX, SEB, PLAGUE, QVF, RICIN, SPOX, TUL	GA
Mass	Kg	> 0 to 100,000	100.0

Table 7. Chemical Weapon Inputs

Inputs	Units	Limits	Defaults
Damage	N/A	Light, Moderate, Severe, Total	Severe
Agent	N/A	ac, acetocya, acrolein, acryloni, allylalc, allylam, allylch, ammonia_liq, ammonia, anth_dry, anth_wet, arsine, bg_dry, bg_wet, bortribr, bortricl, bortrifl, botx_dry, botx_wet, bt_dry, bt_wet, carbonsu, cg, chlorfrm, chloroac, chloroni, chlorsul, ck, cl2_liq, cl2, cn, crbnmnxd, crotonhy, cs, css, dep, diborane, diisprplmn, diketene, dimethhz, dimethsu, ethdibr, ethyloxi, fluorine, formalde, ga, gb, gd, gf, hbr, hcl, hcn, hd, hs, hse, ironpetl, lewisite, malathion, methane, methclfo, mthisocn, mthsulcl, mthybmde, mthyclsi, mthylmcp, mtylhzne, nbutiso, nitrdiox, nitricac, parathion, phos5fl, phosgene, phosphin, phostricl, phsoxycl, plague_dry, plague_wet, propane, qvf_dry, qvf_wet, ricin_dry, ricin_wet, seb_dry, seb_wet, selehexa, sf6, silitetr, spox_dry, spox_wet, stibine, styrene, sulfacid, sulfcl, sulfdxde, tellhexa, tep, toldiiso, toluene, trctylcd, trifchrd, tul_dry, tul_wet, tunghexa, vee_dry, vx	ac
Mass	Kg	0 to 100,000,000	1000

Table 8. Chemical/Biological Facilities Inputs

Weapon	Delivery	GA	GB	GD	GF	HD	VX	TGA	TGD	TGF	THD	TVX	ANTH	BOTX	SEB	PLAGUE	QVF	RICIN	SPOX	TUL
100kg Bomb	1 AC* 2 Drops* 4 Bombs	X	X	X	X	X	X	X	X	X	X	X								
	2 AC* 8 Bombs	X	X	X	X	X	X	X	X	X	X	X								
	4 AC* 8 Bombs	X	X	X	X	X	X	X	X	X	X	X								
250kg Bomb	1 AC* 2 Drops* 4 Bombs	X	X	X	X	X	X	X	X	X	X	X								
	2 AC* 8 Bombs	X	X	X	X	X	X	X	X	X	X	X								
	4 AC* 8 Bombs	X	X	X	X	X	X	X	X	X	X	X								
500kg Bomb	1 AC* 2 Drops* 4 Bombs	X	X	X	X	X	X	X	X	X	X	X								
	2 AC* 8 Bombs	X	X	X	X	X	X	X	X	X	X	X								
	4 AC* 8 Bombs	X	X	X	X	X	X	X	X	X	X	X								
Rocket	1 AC* 48 Rockets	X	X	X	X	X	X	X	X	X	X	X								
	2 AC* 48 Rockets	X	X	X	X	X	X	X	X	X	X	X								
Aerial Spray	Aerial Spray												X	X	X	X	X	X	X	X
152mm Tube	Battery (tube)	X	X	X	X	X	X	X	X	X	X	X								
	Battalion (tube)	X	X	X	X	X	X	X	X	X	X	X								
122mm Rocket	Battery (rocket)	X	X	X	X	X	X	X	X	X	X	X								
	Battalion (rocket)	X	X	X	X	X	X	X	X	X	X	X								
120mm Mortar	Mortar 72 Rounds					X														
	Mortar 36 Rounds					X														
Missile Bulk	Missile Bulk	X	X	X	X	X	X	X	X	X	X	X								
Missile Subs	Missile Subs	X	X	X	X	X	X						X	X	X	X	X	X	X	X
Ground Spray	Ground Spray	X	X	X	X	X	X													
Land Mine	Land Mine					X	X													

Table 9. CBwpn Munition/Delivery/Agent Matrix

Inputs	***STR / STK***	***Units***	***Limits***	***Defaults***
Strike File Type	N/A	N/A	STR, STK	STR
Placement	STK	N/A	Standard, Buried, Contained	Standard
Yield	STR / STK	KT	> 0	10
HOB	STR / STK	feet		0
PA	STR	%	0 to 100	0
CEP	STR	feet	> 0	0
FF	STR / STK	N/A	0.0 to 1.0	1.0
Delfic Rise	N/A	N/A	TRUE, FALSE	TRUE

Table 10. Nuclear Weapons Input

Inputs	Units	Limits	Defaults
Name	N/A	50 characters	
Plot Type	N/A	Default, Surface Dosage, Probability of Casualty, Probability of Mortality, Casualty Table	Default
Agent	N/A	Select from agents involved in existing CB incidents. Not applicable for Plot Type = Default.	
Time	N/A	Last	Last

Table 11. CB-type Plot Inputs

Inputs	Units	Limits	Defaults
Name	N/A	50 characters	
Plot Type	N/A	Blast Circle, Prompt Circle, Thermal Circle, Probability of Casualty, Probability of Fatality, Radiation Dose	Blast Circle
Time	N/A	Last	Last

Table 12. Nuclear-type Plot Inputs

Inputs	Units	Limits	Defaults
Name	N/A	50 characters	
Type Of Points	N/A	Stationary, Moving	Stationary
Type Of Track	N/A	GND, AIR, SUB	GND
Threat	N/A	UNK, FRD, HOS, NEU	UNK
Comments	N/A	N/A	

Table 13. Tracks Input

Inputs	Units	Limits	Defaults
Coordinate Units	N/A	Geodetic (ddd.ssss), Geodetic (ddd mm ss.ss), UTM, MGRS	Geodetic (ddd.ssss)
Latitude	dd.ssss, dd mm ss.ss	90.0000S to 90.0000N or, 90 00 00.00S to 90 00 00.00N	0.0000 N or 0 00 00.00N
Longitude	ddd.ssss, ddd mm ss.ss	180.0000E to 180.0000W or, 180 00 00.00S to 180 00 00.00W	0.0000 E or 0 00 00.00E
Elevation	feet	0 to 100,000	0
Datum	N/A	See list in	WGS-84

Table 14. Geodetic Location Inputs

Inputs	Units	Limits	Defaults
UTM Zone	N/A	1 to 60	31
UTM Hemisphere	N/A	N, S	N
UTM Easting	meters	100,000 to 900,000	166021
UTM Northing	meters	0 to 10,000,000	0
Elevation	feet	0 to 100,000	0
Datum	N/A	See list in	WGS-84

Table 15. UTM Location Inputs

Inputs	Units	Limits	Defaults
MGRS Zone	N/A	1-60	31
MGRS Zone Letter	N/A	A-Z (except I and O)	N
MGRS Grid Designator	N/A	AA-ZZ	AA
MGRS Easting	meters	0 to 100,000	66021
MGRS Northing	meters	0 to 100,000	00000
Elevation	feet	0 to 100,000	0
Datum	N/A	See list in	WGS-84

Table 16. MGRS Location Inputs

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. CSCI BY IWMDT VERSION

CSCI	OPR	Configuration Coordinator or Equivalent
Munitions Effects Assessment (MEA) 5.0	Defense Threat Reduction Agency DTRA/TDOS 6801 Telegraph Rd. Alexandria, VA 22310	DTRA/TDOS LTC William Harman 6801 Telegraph Rd. Alexandria, VA 22310 Phone: (703) 325-7856 Email: William.Harman@DTRA.mil
Hazard Prediction and Assessment Capability (HPAC) 4.0.3	Defense Threat Reduction Agency DTRA/TDOC 6801 Telegraph Rd. Alexandria, VA 22310	DTRA/TDOC Ron Meris 6801 Telegraph Rd. Alexandria, VA 22310 Phone: (703) 325-0608 Email: Ron.Meris@DTRA.mil
Integrated Nuclear Computational Aid (INCA)	Defense Threat Reduction Agency DTRA/TDNE 6801 Telegraph Rd. Alexandria, VA 22310	DTRA/TDNE Gene Stokes 6801 Telegraph Rd. Alexandria, VA 22310 Phone: (703) 325-6414 Email: Eugene.Stokes@DTRA.mil
Integrated Weapons of Mass Destruction Toolset (IWMDT) framework	Defense Threat Reduction Agency DTRA/TDOI 6801 Telegraph Rd. Alexandria, VA 22310	DTRA/TDOI Dave Myers 6801 Telegraph Rd. Alexandria, VA 22310 Phone: (703) 325-6883 Email: Todd.Hann@DTRA.mil

Table A-1 CSCIs for the IWMDT Prototype

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Major Ric Jones
   6801 Telegraph Road
   Alexandria, VA 22030

4. Dr. Man-Tak Shing
   Computer Science Department
   Naval Postgraduate School
   Monterey, California

5. Dr. Doron Drusinsky
   Computer Science Department
   Naval Postgraduate School
   Monterey, California

6. Dr. Peter Denning
   Computer Science Department
   Naval Postgraduate School
   Monterey, California